Proyecto Fin de Máster en Ingeniería de Computadores

Máster en Investigación en Informática
Facultad de Informática
Universidad Complutense de Madrid

IMPACTO DE LA SEGURIDAD EN REDES INALÁMBRICAS DE SENSORES IEEE 802.15.4

Autor:
Carlos García Arano
Directores:
David Atienza Alonso
Joaquín Recas Piorno

CURSO ACADÉMICO 2009/2010

AUTORIZACIÓN

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Impacto de la seguridad en Redes Inalámbricas de Sensores IEEE 802.15.4", realizado durante el curso académico 2009-2010 bajo la dirección de David Atienza Alonso y Joaquín Recas Piorno en el Departamento de Arquitectura de Computadores y Automática (DACYA), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

"*If you think you can solve your security problems, then you don't understand the problems and you don't understand the technology.*"

Bruce Schneier

AGRADECIMIENTOS

Quisiera agradecer a mis padres el haberme proporcionado una educación.

Gracias también a Noelia Morón, que me impulsó a realizar el Máster y me ayudo con sus amplios conocimientos.

Agradecimientos a David Pérez, por indicarme los primeros pasos con las redes de sensores, y a David Gascón, de Libelium, por permitirme hacer uso de parte de su trabajo en este proyecto.

Y no puedo dejar de incluir a Pablo Ambrosy, que sin su apoyo y comprensión no habría podido plantearme siquiera realizar el Máster de Investigación Informática.

RESUMEN

Este proyecto analiza el impacto que provoca el uso de las funcionalidades de seguridad descritas en el estándar IEEE 802.15.4 en las redes de sensores. Se han implementado los diferentes niveles de seguridad propuestos con la asistencia de un módulo hardware criptográfico y se han integrado en un desarrollo basado en FreeRTOS de la capa MAC del estándar. Estas funcionalidades permiten garantizar la confidencialidad e integridad de las comunicaciones, pero suponen un coste en diferentes aspectos que se ha de cuantificar. Se han analizado los costes en el consumo de energía, ya que los recursos energéticos de los sensores son una de las principales limitaciones de este tipo de redes. Los resultados evidencian un aumento en el consumo, pero casi exclusivamente derivados de incremento del tiempo de transmisión.

ABSTRACT

This project analyzes the impact arising from the use of security features described in the IEEE 802.15.4 standard in wireless sensors networks. The proposed security levels have been implemented using the available hardware support, and integrated in a FreeRTOS based MAC layer. These functionalities help ensuring confidentiality and integrity of communications, but mean an overhead in some aspects that should be measured. Energy consumption has been analyzed, as it is one of the main limitations of these kinds of networks. Results show an increase in power consumption, but it is due, almost solely, to the increase of the transmission time.

PALABRAS CLAVE

IEEE 802.145.4, ZigBee, redes inalámbricas de sensores, seguridad, AES, consumo de energía

KEY WORDS

IEEE 802.145.4, ZigBee, wireless sensors networks, security, AES, energy consumption

ÍNDICE

INTRODUCCIÓN 1
Objetivos 1
Estructura del documento 1
1. REDES DE SENSORES 3
1.1. Concepto de Red de Sensores 3
1.2. Historia de las Redes de Sensores 3
1.3. Características de una Red de Sensores 4
1.4. Aplicaciones 6
2. IEEE 802.15.4 9
2.1. Introducción al estándar 9
2.2. Componentes Básicos 10
2.3. Capa física del 802.15.4 11
2.4. Capa MAC del 802.15.4 12
2.4.1. Funcionamiento 13
2.4.2. Mecanismos de Robustez 14
2.4.3. Implementación usada durante el proyecto 14
3. SEGURIDAD DE LA INFORMACIÓN 15
3.1. Concepto de Seguridad de la Información 15
3.1.1. Confidencialidad 15
3.1.2. Integridad 16
3.1.3. Disponibilidad 16
3.2. Riesgos, amenazas y Vulnerabilidades en Redes de Sensores 17
3.3. Vectores de Ataque 17
3.3.1. Capa Física 18
3.3.2. Capa MAC 19
3.3.3. Capa de Red 19
3.3.4. Capa de aplicación 21
4. SEGURIDAD EN IEEE 802.15.4 23
4.1. Descripción 23
4.2. Formatos de trama 23
4.3. Niveles de seguridad 25
4.3.1. Advanced Encryption Standard 25
4.3.2. CBC-MAC 26
4.3.3. CTR 27
4.3.4. CCM 28
5. ENTORNO DE TRABAJO 29
5.1. Shimmer 29
5.1.1. MPS430 29
5.1.2. Radio 30
5.2. HURRAY y TinyOS 30
5.2.1. TinyOS y nesC 30
5.2.2. HURRAY 32
5.3. FreeRTOS 32
6. FUNCIONALIDADES IMPLEMENTADAS 35
6.1. Soporte Hardware 35
6.1.1. Claves 36
6.1.2. Vectores de inicialización 36
6.1.3. Modos de operación 37
6.2. Niveles de seguridad 37
6.2.1. CBC-MAC 37
6.2.2. CTR 38
6.2.3. CCM 38
6.2.4. Standalone 39
6.3. Procedimiento de medida 39

6.3.1. Descripción ... 39
6.3.2. Funcionamiento del nodo coordinador ... 39
6.3.3. Funcionamiento del nodo sensor ... 40
6.4. Pruebas realizadas y resultados ... 40
6.4.1. Análisis de consumo ... 42
6.4.2. Análisis del tiempo ... 43
6.4.3. Análisis del tamaño útil ... 44
6.4.4. Cuadrantes ... 45
7. CONCLUSIONES ... 47
7.1. Conclusiones del estudio ... 47
7.2. Lineas futuras ... 47
APÉNDICE ... 49
A. Energía consumida ... 49
B. Tiempo de transmisión ... 49
C. Tamaño de trama ... 50
D. Medidas ... 51
E. Cálculos intermedios ... 52
BIBLIOGRAFÍA ... 53

ÍNDICE DE FIGURAS

Figura 1 Kit de desarrollo de Shimmer ... 3
Figura 2 Estructura de una red de sensores ... 5
Figura 3 Estructura de un sensor ... 5
Figura 4 Aplicaciones potenciales de las redes de sensores ... 7
Figura 5 Topología en estrella ... 10
Figura 6 Topología en malla ... 11
Figura 7 Topología Cluster Tree ... 11
Figura 8 Banda de 2,4 GHz ... 12
Figura 9 PDU de la capa PHY ... 12
Figura 10 Uso de *beacons* ... 13
Figura 11 Uso de *beacons* y GTS ... 14
Figura 12 Comunicación básica ... 15
Figura 13 Compromiso de la confidencialidad ... 15
Figura 14 Alteración de la información ... 16
Figura 15 Inyección de información ... 16
Figura 16 Denegación de servicio ... 17
Figura 17 Pila OSI ... 18
Figura 18 Ataque Sybil ... 19
Figura 19 Sinkhole ... 20
Figura 20 Wormhole ... 20
Figura 21 HELLO Flood ... 21
Figura 22 Cabecera IEEE 802.15.4 ... 24
Figura 23 Subcampos de ASH ... 24
Figura 24 AES-CBC-MAC ... 26
Figura 25 AES-CTR ... 27
Figura 26 AES-CCM ... 28
Figura 27 Anverso del Shimmer ... 29
Figura 28 Reverso del Shimmer ... 29
Figura 29 Formato de IV ... 36
Figura 30 Circuito de pruebas ... 39
Figura 31 Diseño de las pruebas ... 40
Figura 32 Captura de tramas ... 40
Figura 33 Lecturas de tensión V_R sin seguridad ... 41
Figura 34 Lecturas de tensión V_R con cifrado (CTR) ... 41
Figura 35 Consumo de energía ... 42

Figura 36 Impacto en el consumo 42
Figura 37 Tiempos de transmisión 43
Figura 38 Impacto en el tiempo 44
Figura 39 Impacto en el tamaño de trama 44
Figura 40 Cuadrante para tamaños de 12 bytes 45
Figura 41 Cuadrante para tamaños de 96 bytes 46

ÍNDICE DE TABLAS

Tabla 1 Comparativa de estándares RF 9
Tabla 2 Frecuencias disponibles 12
Tabla 3 Niveles de seguridad 25
Tabla 4 Energía consumida 49
Tabla 5 Porcentaje de consumo 49
Tabla 6 Tiempo de transmisión 50
Tabla 7 Porcentaje del tiempo de transmisión 50
Tabla 8 Porcentaje del tamaño de trama 50
Tabla 9 Medidas de las pruebas 51
Tabla 10 Cálculos intermedios 52

INTRODUCCIÓN

En este primer capítulo se describirán brevemente los objetivos principales del proyecto y la estructura de este documento.

OBJETIVOS

Muchas de las aplicaciones de las redes de sensores tratan información sensible que debe ser protegida para evitar su difusión, así como información que es crítica para el correcto funcionamiento de la red. Por ese motivo, es necesario asegurar la identidad de los diferentes dispositivos que conforman la red, evitando la asociación de nodos ajenos, y establecer enlaces de comunicación confidenciales entre los interlocutores. Para ello, el estándar de comunicaciones IEEE 802.15.4 describe una serie de mecanismos, basados en la criptografía, que permiten garantizar los requisitos básicos de una comunicación segura. Concretamente ofrece tres modos diferenciados: CBC-MAC, que aporta autenticación e integridad, CTR, que proporciona confidencialidad, y CCM que combina los dos anteriores para garantizar la privacidad, la integridad y la autenticación.

Debido a las restricciones propias de los dispositivos que conforman la red, estos mecanismos deben ofrecer un compromiso aceptable entre la seguridad y el consumo de energía y de recursos *hardware*. Este proyecto pretende analizar estos dos aspectos. Por un lado, identificar las principales amenazas a la seguridad en redes de sensores, presentar las operaciones de seguridad que define el estándar, y analizar cómo protegen la información. Por otro lado, determinar el impacto que produce el uso de estas operaciones en los dispositivos, calculando el consumo de energía, de ancho de banda y de tiempo de transmisión, determinando las causas.

El análisis de consumo es resuelto en muchos artículos previos mediante estimaciones o simulaciones, o realizando las operaciones criptográficas por *software*. En este proyecto se ha desarrollado una capa de seguridad totalmente funcional, que permite la toma de datos sobre una plataforma comercial de bajo coste, utilizando el soporte *hardware* de un transceptor radio ampliamente difundido y con aplicaciones reales de adquisición de datos. Adicionalmente, se ha optado por la utilización de un sistema operativo en tiempo real, lo que en sí mismo ya es una innovación, pues la mayoría de las implementaciones están basadas en sistemas operativos basados en eventos. A día de hoy no se ha encontrado[1] ninguna implementación *open source* con soporte *hardware* para las operaciones criptográficas.

Debido a esto, un objetivo secundario del proyecto es dotar al Departamento de una plataforma que integre las características de seguridad para realizar los estudios y mediciones que se enmarcan en el proyecto auspiciado por la UCM y la EPFL.

ESTRUCTURA DEL DOCUMENTO

Estableciendo como base del proyecto los temas que se acaban de exponer en el apartado anterior, el proyecto está compuesto por 7 capítulos. El primer capítulo (Capítulo 0) es un capítulo introductorio que sirve al lector para conocer los objetivos y estructura del documento. Los capítulos siguientes (Capítulo de 1 al 6) conforman el grueso del proyecto y es aquí donde se reflejan los resultados obtenidos durante el mismo. Para finalizar, en el último capítulo (Capítulo 6) se exponen las conclusiones obtenidas durante la realización del proyecto, así como las posibles líneas a seguir en el futuro.

Seguidamente se describen brevemente cada uno de los capítulos que componen este documento:

- **CAPÍTULO 0. Introducción**
  Descripción de los objetivos principales del proyecto y la estructura del documento.

1 Los siguientes trabajos (MAC API, PixieMAC, MACdongle, open-ZB (HURRAY), MeshNetics, OpenMAC, FreakZ Open Source Zigbee Stack) carecen de soporte *hardware*, algunos ni siquiera soportan la seguridad de IEEE 802.15.4. Igualmente, es complicado encontrar implementaciones reales en publicaciones académicas.

- **CAPÍTULO 1. Redes de sensores**
  Breve introducción a las redes de sensores, lo que son, su historia, sus características y sus posibles usos en diferentes ámbitos.

- **CAPITULO 2. IEEE 802.15.4**
  Descripción general del estándar IEEE 802.15.4. Se detallarán tanto las características más relevantes como los distintos modos de funcionamiento que dispone, haciendo especial hincapié en el tipo de transceptor utilizado durante el proyecto.

- **CAPÍTULO 3. Seguridad de la Información**
  Introducción a los conceptos que definen la seguridad y análisis de los factores que afectan a las redes de sensores.

- **CAPÍTULO 4. Seguridad en IEEE 802.15.4**
  Descripción de las características de seguridad que especifica el estándar del IEEE, analizando su aportación a los conceptos definidos en el apartado anterior.

- **CAPÍTULO 5. Entono de trabajo**
  Descripción y análisis de los componentes, tanto hardware (plataforma Shimmer) como software (TinyOS, NesC, Hurray, FreeRTOS), que han sido necesarios en algún momento para llevar a cabo el presente proyecto.

- **CAPITULO 6. Funcionalidades Implementadas**
  Descripción del trabajo realizado para poder llevar a cabo el análisis del objetivo del proyecto. Se detalla el procedimiento de medida y las pruebas realizadas, y se presentan los resultados obtenidos.

- **CAPÍTULO 7. Conclusiones**
  Se exponen las conclusiones obtenidas durante la realización de este estudio, así como las posibles líneas a seguir en el futuro.

1. REDES DE SENSORES

En este primer capítulo haremos una introducción a las redes de sensores, lo que son, su historia, sus características y sus posibles usos en diferentes ámbitos.

1.1. CONCEPTO DE RED DE SENSORES

Las redes de sensores están formadas por un grupo de sensores con ciertas capacidades sensitivas y de comunicación, las cuales permiten formar redes inalámbricas *ad hoc* sin infraestructura física preestablecida ni administración central. Las redes de sensores es un concepto relativamente nuevo en adquisición y tratamiento de datos con múltiples aplicaciones en distintos campos tales como entornos industriales, domótica, entornos militares, detección ambiental y medicina. Esta clase de redes se caracterizan por su facilidad de despliegue y por ser autoconfigurables, pudiendo convertirse en todo momento en emisor y receptor, ofrecer servicios de encaminamiento entre nodos sin visión directa, así como registrar datos referentes a los sensores locales de cada nodo. Actualmente existen diversas líneas de investigación centradas en desarrollar mecanismos que permitan una gestión eficiente de la energía y que, por tanto, permitan gozar a las redes de sensores de una alta tasa de autonomía que las hagan plenamente operativas

Cada nodo, como ente individual de una red de sensores, no deja de ser un pequeño ordenador, con un pequeño procesador, una memoria de programa y una memoria para almacenar variables, pero al que también agregamos unos pequeños periféricos I/O (entrada/salida) tales como un transceptor radio y un conversor analógico/digital, utilizado para la adquisición de los datos de los sensores locales. En concreto, los sensores con los que trabajamos en este proyecto poseen un procesador MSP430 a 8Mhz con 48 KB de memoria de programa (Flash) y 10 KB de memoria volátil (RAM), con el transceptor CC2420 de Texas Instruments, que soporta IEEE 802.15.4 [1]

Figura 1 Kit de desarrollo de Shimmer

1.2. HISTORIA DE LAS REDES DE SENSORES

Como sucede con muchas tecnologías, el desarrollo de las redes de sensores nació en el seno de la investigación para aplicaciones militares. La primera de estas redes, conocida con el nombre de SOSUS (*Sound Surveillance System*) [2], fue desarrollada por Estados Unidos durante la guerra fría y se trataba de una red de sensores acústicos desplegados en el fondo del mar cuya misión era desvelar la posición de los silenciosos submarinos soviéticos. Paralelamente a ésta, también en EEUU, se desplegó una red de radares aéreos a modo de sensores que han ido evolucionando hasta dar lugar a los famosos aviones AWACS, que no son más que sensores aéreos. Mientras que SOSUS ha evolucionado hacia aplicaciones civiles como control sísmico y biológico, AWACS sigue teniendo un papel activo en las campañas de guerra.

Estas primeras redes de sensores generalmente adoptaban una estructura de procesado jerárquico donde el procesado ocurría en niveles consecutivos antes de que la información sobre los eventos de interés llegase al usuario. Los nodos eran grandes estaciones distantes espacialmente y su comunicación tenía lugar a través de una infraestructura cableada.

El nacimiento de la investigación moderna podría atribuirse al programa DSN (*Distributed Sensor Networks*) de la DARPA que comenzó a partir de 1980 [2], en el que se pretendía comprobar si el método de comunicación de la recién aparecida Arpanet era extensible a las redes de sensores. La red de pruebas estaba compuesta por muchos nodos sensores de bajo coste distribuidos espacialmente. Los nodos colaboraban unos con otros pero operaban de forma autónoma, y la información se enrutaba hacia cualquier nodo que pudiese hacer el mejor uso de la información. Gracias al proyecto DNS, por tanto, se crearon los primeros sistemas operativos (Accent) y lenguajes de programación (SPLICE) orientados de forma específica a las redes de sensores, lo que permitió dar lugar a nuevos sistemas militares como CEC (*Cooperative Engagement Capability*). Dicho sistema militar consiste básicamente en un grupo de radares que comparten toda su información con el fin de obtener un mapa común con una mayor exactitud y precisión.

Aunque los primeros investigadores en redes de sensores tenían en mente redes compuestas por un gran número de pequeños sensores, la tecnología para pequeños sensores todavía no estaba suficientemente desarrollada ya que aun no se satisfacían algunos requisitos de gran importancia en este tipo de redes tales como la autonomía y el tamaño.

En las décadas de los 80 y los 90, este tipo de redes se convirtieron en un componente crucial de los sistemas militares. Las redes de sensores perfeccionaban el rendimiento de la detección y el rastreo a través de múltiples observaciones, aprovechando su amplio rango de detección y su pequeño tiempo de respuesta, como por ejemplo en sistemas de detección de francotiradores [3]. En esta década destacan programas como SensIt, una vez más desarrollado por la DARPA que perseguía dos objetivos claves. Por un lado desarrollar nuevas técnicas de operación en red apropiadas para entornos ad hoc muy dinámicos y por otro lado, desarrollar un método para la extracción de información actualizada, útil y fiable desde la red de sensores desplegada.

Avances recientes en computación y comunicaciones han causado un cambio significativo en la investigación en redes de sensores, acercándola hacia la consecución de la visión original. Baratos y diminutos sensores basados en la tecnología de sistemas microelectromecánicos (MEMS), la comunicación inalámbrica y procesadores baratos de bajo consumo, permiten el despliegue de redes inalámbricas *ad hoc* para multitud de aplicaciones, ya sea para la monitorización del entorno, monitorización de seguridad o *tracking.*

1.3. CARACTERÍSTICAS DE UNA RED DE SENSORES

El desarrollo de las redes de sensores requiere tecnologías de tres áreas de investigación diferentes: detección, comunicación, y computación (incluyendo *hardware*, *software* y algoritmia).

Los nodos sensores se encuentran normalmente esparcidos en un campo sensor (ver Figura 2). Cada uno de estos nodos sensores esparcidos por la red tiene capacidad tanto para recolectar datos, como para enrutarlos hacia el nodo recolector (*sink node*) mediante una arquitectura *ad hoc* de múltiples saltos.

El nodo recolector puede comunicarse con el nodo administrador (gestor de tareas) vía Internet, vía satélite o de forma directa.

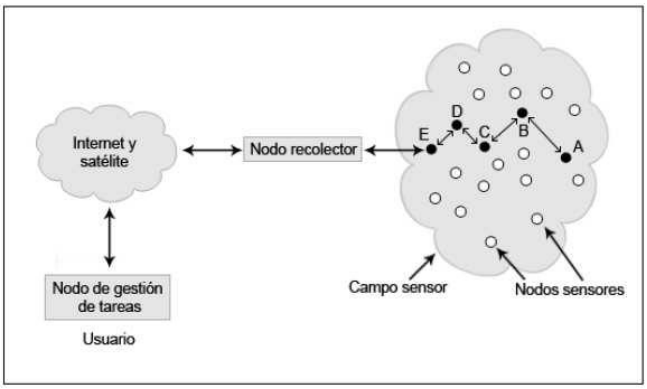

Figura 2 Estructura de una red de sensores

El diseño de una red de sensores como la descrita aquí está altamente influenciado por los siguientes factores [4]:

- **Tolerancia a fallos**: Algunos nodos sensores pueden fallar o bloquearse debido a la falta de energía, o recibir daños físicos o interferencias medioambientales. El fallo de nodos sensores no debería comprometer el funcionamiento global de la red sensora. Este es el principio de la tolerancia a fallos o fiabilidad.

- **Escalabilidad**: Los nuevos diseños deben ser capaces de trabajar con un número de nodos del orden de centenares, millares, e incluso, dependiendo de la aplicación, millones. También deben tener en cuenta la alta densidad, que puede llegar hasta algunos centenares de nodos sensores en una región, que puede ser menor de 10 metros de diámetro.

- **Costes de producción**: Dado que las redes de sensores consisten en un gran número de nodos sensores, el coste de un nodo individual es clave para que una red inalámbrica sea rentable en comparación con una cableada. Si el coste de la red es más caro que el despliegue de sensores tradicionales, la red sensora no está justificada desde el punto de vista económico.

- **Limitaciones hardware**: Un nodo sensor está constituido por cuatro componentes básicos, como muestra la Figura 3 Estructura de un sensor: una unidad sensora, una unidad de proceso, una unidad transceptora, y una unidad de energía, aunque pueden tener también componentes adicionales dependiendo de su aplicación como un sistema de localización, un generador de energía o un movilizador.

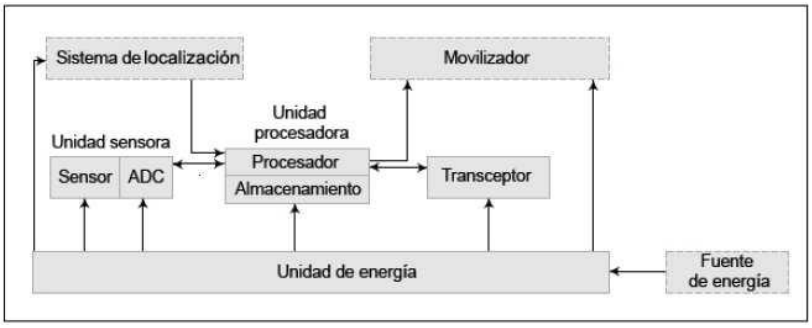

Figura 3 Estructura de un sensor

Las señales analógicas producidas por los sensores, basadas obviamente en el fenómeno observado, son convertidas a señales digitales por el conversor ADC, para ser pasadas después a la unidad de proceso.

La unidad de proceso, generalmente asociada a una pequeña unidad de almacenamiento, maneja los procedimientos necesarios para que el nodo sensor colabore con los demás en la realización de las tareas de percepción asignadas. Una unidad transceptora conecta el nodo a la red.

Uno de los componentes más importantes de un nodo sensor es la fuente de alimentación. La fuente de alimentación puede ser abastecida por unidades de captadoras de energía como es el caso de las células solares.

- **Topología**: El despliegue de un gran número de nodos densamente distribuidos precisa de un mantenimiento y gestión de la topología cuidadosos. Se pueden dividir las tareas de mantenimiento y cambio de la topología en tres fases:
    - Pre-despliegue y despliegue: Los nodos sensores pueden ser arrojados en masa o colocados uno por uno en el campo sensor.
    - Post-despliegue: Después del despliegue, los cambios de topología son debidos a cambios en la posición de los nodos sensores, accesibilidad (debido a interferencias intencionadas (jamming), ruido, obstáculos móviles, etc), energía disponible, funcionamiento defectuoso y detalles de las tareas encomendadas.
    - Despliegue de nodos adicionales: Nodos sensores adicionales pueden ser desplegados en cualquier momento para reemplazar nodos defectuosos o debido a cambios en la dinámica de las tareas.

- **Entorno**: Los nodos sensores son desplegados densamente bien muy cerca o directamente en el interior del fenómeno a ser observado. Por consiguiente, normalmente trabajan desatendidos en áreas geográficas remotas. Pueden estar trabajando en el interior de maquinaria grande, en el fondo del océano, en un área contaminada biológicamente o químicamente, en un campo de batalla más allá de las líneas enemigas, así como en edificios y hogares.

- **Medio de transmisión**: En una red de sensores multisalto, los nodos de comunicaciones están conectados mediante un medio inalámbrico. Estas conexiones pueden estar formadas por medios radio, infrarrojo o óptico, aunque la gran mayoría del hardware actual para redes de sensores está basado en RF.

  Otro posible modo de comunicación entre nodos en redes de sensores es mediante infrarrojos. La comunicación por infrarrojos no necesita licencia y es robusta frente a interferencias producidas por dispositivos eléctricos. Los transceptores basados en infrarrojos son baratos y fáciles de construir.

  Otro desarrollo interesante es el del Smart Dust, que es un sistema autónomo de percepción, computación y comunicación que utiliza el medio óptico para transmitir.

  Ambos medios, infrarrojo y óptico, requieren de visión directa entre el nodo o nodos transmisores y receptores.

- **Consumo energético**: Los nodos sensores inalámbricos, por lo general, están equipados con una fuente energética limitada (< 0,5 Ah, 1.2 V). En los escenarios de algunas aplicaciones, la recarga de los recursos energéticos puede ser imposible. El tiempo de vida de los nodos sensores, en consecuencia, muestra una gran dependencia del tiempo de vida de la batería.

  En una red sensora ad hoc multisalto, cada nodo desempeña el doble rol de origen de información y enrutador de información.
  El funcionamiento defectuoso de algunos nodos puede causar cambios de topología significativos y puede requerir re-enrutamiento de los paquetes y reorganización de la red. De aquí que, la conservación y administración energética tomen una importancia adicional.

1.4. APLICACIONES

Dentro del campo de las redes móviles *ad hoc*, las redes de sensores son las que parecen tener un futuro más prometedor.

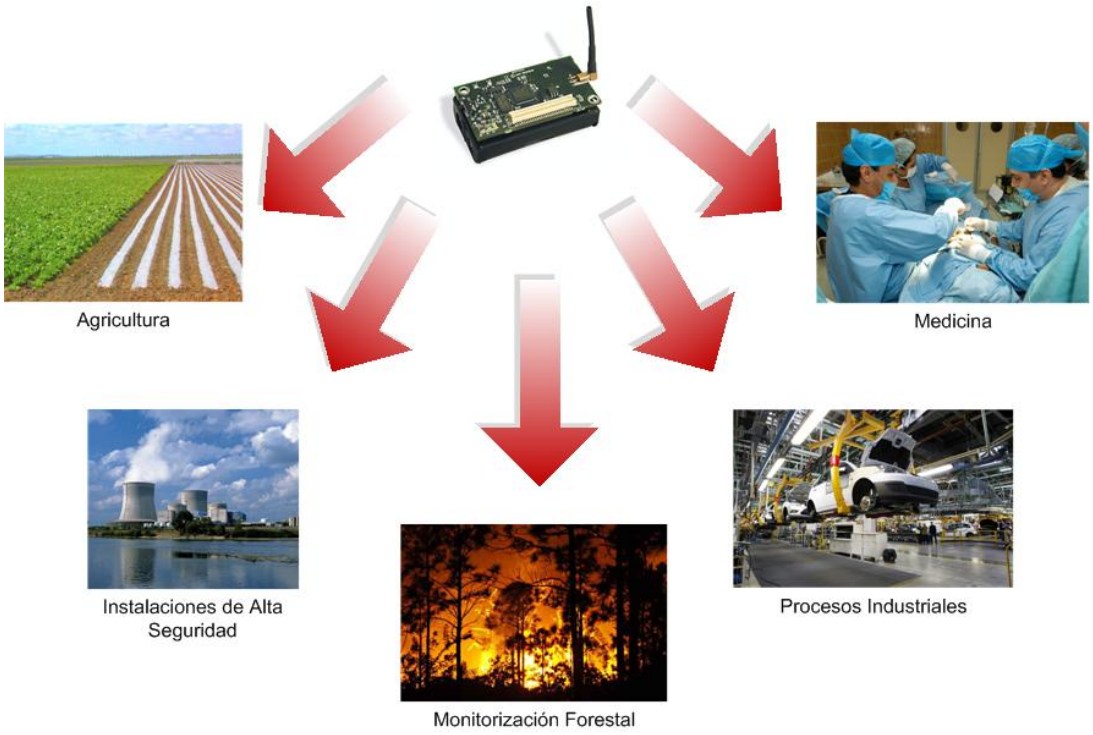

Figura 4 Aplicaciones potenciales de las redes de sensores

Pasando de largo las aplicaciones militares que antes hemos comentado en la historia de las redes de sensores, éstas tienen usos civiles interesantes como podemos ver en la figura anterior y descritos también a continuación:

- **Entornos de alta seguridad**: Existen lugares que requieren altos niveles de seguridad, por ejemplo para evitar ataques terroristas, tales como centrales nucleares, aeropuertos, edificios del gobierno de paso restringido. Aquí gracias a una red de sensores se pueden detectar situaciones que con una simple cámara sería imposible.

- **Sensores ambientales**: El control ambiental de vastas áreas de bosque o de océano, sería imposible sin las redes de sensores. El control de múltiples variables, como temperatura, humedad, fuego, actividad sísmica así como otras. También ayudan a expertos a diagnosticar o prevenir un problema o urgencia y además minimizar el impacto ambiental del presencia humana.

- **Sensores industriales**: Dentro de fábricas existen complejos sistemas de control de calidad, el tamaño de estos sensores les permite estar allí donde se requiera.

- **Automoción**: Las redes de sensores son el complemento ideal a las cámaras de tráfico, ya que pueden informar de la situación del tráfico en ángulos muertos que no cubren las cámaras y también pueden informar a conductores de la situación, en caso de atasco o accidente, con lo que estos tienen capacidad de reacción para tomar rutas alternativas.

- **Medicina**: Es otro campo bastante prometedor. Con la reducción de tamaño que están sufriendo los nodos sensores, la calidad de vida de pacientes que tengan que tener controlada sus constantes vitales (pulsaciones, presión, nivel de azúcar en sangre, etc), podrá mejorar sustancialmente. En estos entornos, las posibles aplicaciones están limitadas por la compatibilidad electromagnética con el material hospitalario.

- **Domótica**: Su tamaño, economía y velocidad de despliegue, la hacen una tecnología ideal para *domotizar* el hogar a un precio asequible.

2. IEEE 802.15.4

En este segundo capítulo se realizará una breve introducción al estándar IEEE 802.15.4 [5, 6] . Se detallarán tanto las características más relevantes como los distintos modos de funcionamiento que dispone, haciendo especial hincapié en el tipo de transceptor utilizado durante el proyecto.

2.1. INTRODUCCIÓN AL ESTÁNDAR

El estándar IEEE 802.15.4 surgió debido a la escasez de estándares inalámbricos de baja tasa de transmisión para redes de sensores. Los estándares disponibles en el mercado (Wi-fi, WiMAX, Bluetooth) estaban orientados hacia aplicaciones con requerimientos de alto ancho de banda como puede ser redes locales, videoconferencia, etc. El inconveniente que surgía al utilizar cualquiera de los estándares antes mencionados era un gran consumo de energía y un gran ancho de banda utilizado frente a las bajas tasas de transmisión y bajos requerimientos de energía necesaria para las redes de sensores. A continuación se puede observar una pequeña comparativa [1] entre 802.15.4 y otros estándares como son Bluetooth, Wi-fi y el estándar ECMA/ISO de UWB (Ultra Wide Band):

Estándares	Ancho de banda	Consumo de potencia	Ventajas	Aplicaciones
Wi-fi	Hasta 54 Mbps	160 mA en reposo	Gran ancho de banda	Navegación por Internet, redes locales, transferencia de ficheros
Bluetooth	1 Mbps	22 mA en reposo	Interoperabilidad, sustituto del cable	Wireless USB, móviles, informática doméstica
IEEE 802.15.4	250 Kbps	3 mA en reposo	Batería de larga duración, bajo coste	Control remoto, productos dependientes de la batería, sensores
UWB	100 Mbps	2 mA en reposo	Gran ancho de banda, bajo consumo	En proceso de estandarización

Tabla 1 Comparativa de estándares RF

En un principio, cada fabricante de nodos sensores optó por utilizar soluciones propietarias, dada la presión ejercida por el mercado, lo que trajo problemas de interoperabilidad entre los diversos fabricantes.

La industria entendió en su momento que hacía falta un nuevo estándar que aunara autonomía, envío de datos de baja capacidad (Kbps) y un bajo coste. Es por tanto con este objetivo por el que nacen tanto el estándar 802.15.4 como Zigbee. En concreto podemos definir Zigbee como una pila de protocolos que permite la comunicación de forma sencilla entre múltiples dispositivos. Zigbee especifica diversas capas, adecuándose al modelo OSI.

Las capas básicas, física y de control de acceso al medio están definidas por el estándar IEEE 802.15.4, LR-WPAN (Low Rate – Wireless Personal Area Network). Este estándar fue diseñado pensando en la sencillez de la implementación y el bajo consumo, sin perder potencia ni posibilidades.

El estándar ZigBee amplía el estándar IEEE 802.15.4 aportando tanto una capa de red que gestiona las tareas de enrutado y de mantenimiento de los nodos de la red, como un entorno de aplicación que proporciona una subcapa que establece una interfaz para la capa de red y los objetos de los dispositivos tanto de ZigBee como del diseñador.

Así pues, los estándares IEEE 802.15.4 y ZigBee se complementan proporcionando una pila completa de protocolos que permiten establecer comunicaciones entre multitud de dispositivos de una forma eficiente y sencilla.

2.2. COMPONENTES BÁSICOS

Dentro del estándar IEEE 802.15.4 se pueden definir dos tipos de nodos según su funcionamiento y la topología utilizada en la red sensorial, que a continuación pasamos a describir brevemente:

- **FFD** (*Full Function Device*): son dispositivos capaces de organizar y coordinar el acceso al medio de otros dispositivos de la misma red. Estos dispositivos se suelen utilizar en redes donde se necesita un nodo central, como puede ser en redes con topología en estrella, y suelen requerir un consumo de energía superior a otros nodos por lo que se suele conectar a la red eléctrica.

- **RFD** (*Reduced Function Device*): son dispositivos con un bajo consumo de energía y de un bajo coste y simplicidad, estos dispositivos se suelen utilizar en cualquier tipo de red.

Dependiendo de la aplicación que se esté desarrollando, se pueden configurar tres tipos de topología: topología en estrella, topología en malla (o *peer-to-peer*) y topología híbrida (o *cluster-tree*). A continuación se definen cada una de ellas:

A. Topología en estrella

En la topología en estrella la comunicación se establece entre los nodos (RFD o FFD) y el nodo central llamado *PAN coordinator*. Una vez se conectan los nodos en una red en estrella, se elige cual va a ser el nodo coordinador de dicha red, el nodo elegido proporciona un identificador de red que no puede ser igual al identificador de otra red dentro del radio de acción de este nodo coordinador (área de cobertura o huella). El nodo coordinador será el que autorice la transmisión a los demás nodos debido a que será este el controlador de la red. Las aplicaciones más comunes que utilizan este tipo de topología son la conexión entre el ordenador personal y los periféricos, domótica o juguetes.

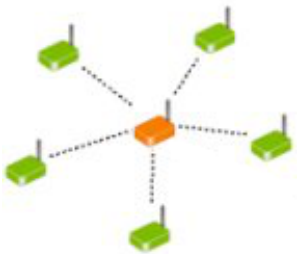

Figura 5 Topología en estrella

B. Topología en Malla o *Peer-to-peer*

En la topología en malla o *peer-to-peer* también existe el papel del dispositivo *PAN coordinator* pero no tiene las mismas funciones relevantes. En contraste con la topología en estrella, cualquier dispositivo puede comunicarse con cualquier otro mientras ambos estén en la misma área de cobertura o utilizando otros nodos para llegar al destino (topología *mesh*) debido a que tienen la misma prioridad a la hora de transmitir. Este tipo de topología es utilizada en redes *ad hoc* y se implementa en distintas aplicaciones como pueden ser control industrial, control de incendios o aplicaciones de inventario. Esta topología permite múltiples saltos entre el nodo origen y destino con lo que conlleva la utilización de protocolos de enrutamiento en este tipo de topología (como por ejemplo, protocolos AODV, DYMO o DSR).

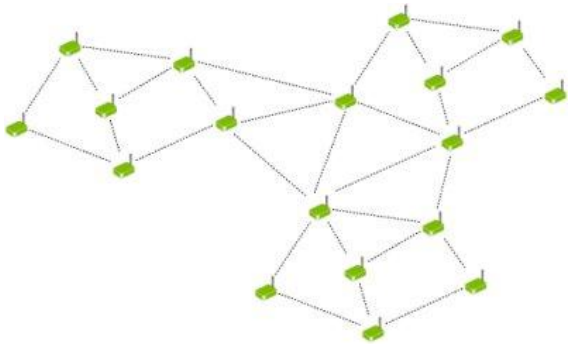

Figura 6 Topología en malla

C. Topología Híbrida o *Cluster-tree*

La topología híbrida o *cluster-tree* es un caso especial de la topología en malla donde se conectan varios dispositivos FFD y RFD entre sí, formando una jerarquía de árbol. En este tipo de topologías existen varios nodos coordinadores en una determinada zona y luego existe el papel del *PAN coordinator* que es el coordinador de toda la red que está en un nivel superior, como se puede observar en la siguiente figura.

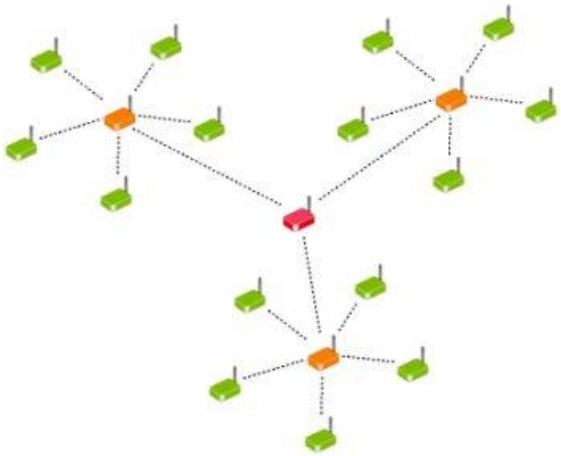

Figura 7 Topología Cluster Tree

2.3. CAPA FÍSICA DEL 802.15.4

La capa física del 802.15.4 está separada en dos subcapas: PHY *data service* y PHY *management* que son las encargadas de transmitir y recibir mensajes a través del medio radio.

Algunas características globales de la capa física son el control del transceptor radio, calidad del enlace (LQI), selección de canal, detector de energía (ED), estimación de la claridad del canal (CCA) para su uso en CSMA-CA a nivel MAC, etcétera.

El estándar define dos opciones de transmisión según la banda de frecuencia utilizada (868/915 MHz y 2450 MHz), ambos basados en el DSSS (*Direct Sequence Spread Spectrum*). DSSS es una técnica de modulación que utiliza un código de pseudorruido para modular directamente una portadora, de tal forma que aumente el ancho de banda de la transmisión y reduzca la densidad de potencia espectral. La señal resultante tiene un espectro muy parecido al del ruido, de tal forma que a todos los radiorreceptores les parecerá ruido menos al que va dirigida la señal.

Como se puede observar en la Tabla 2, se obtienen distintas velocidades de transmisión dependiendo la frecuencia que se utilice, lo que conlleva que a mayor frecuencia, mayor velocidad, pero menor área de cobertura debido a la atenuación de la señal a frecuencias elevadas.

También se puede observar los distintos tipos de modulación que se utilizan dependiendo de la frecuencia utilizada (BPSK y O-QPSK) pero no se va a entrar más en detalle puesto que no es objetivo del proyecto.

MHz	Banda de frecuencia (MHz)	Modulación	Kbps	Ksimbolos/s
868/915	868-868,6 902-928	BPSK BPSK	20 40	20 40
2450	2000-2483,5	O-QPSK	250	62,5

Tabla 2 Frecuencias disponibles

En la banda de 2,4GHz, que es la más eficiente en cuanto al uso del ancho de banda, se dispone de un total de 16 bandas o canales de 2 MHz con una distancia entre canales de 5MHz para evitar interferencias, tal y como vemos en la Figura 8 Banda de 2,4 GHz.

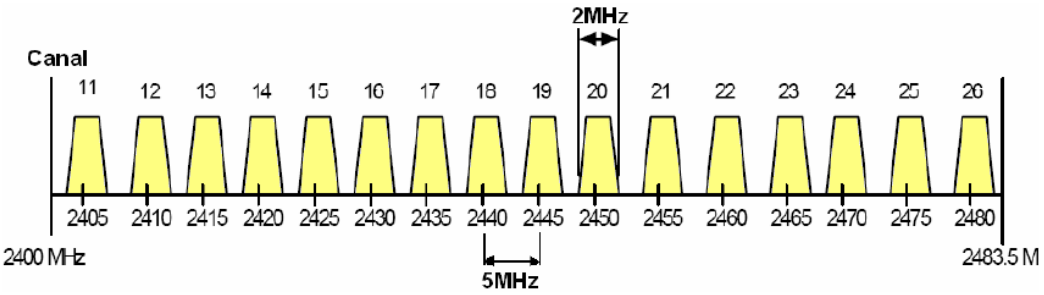

Figura 8 Banda de 2,4 GHz

La unidad de datos de nivel físico (PPDU) que es ilustrada en la Figura 9 está compuesta por tres partes bien definidas:

- SHR (*Synchronization Header*), permite a un dispositivo receptor sincronizarse para poder leer bien la información contenida en la PPDU, también indica el final de trama ya que la trama puede tener una longitud variable.
- PHR (*Physical H*eader) indica la longitud de información ya que ésta puede ser variable como hemos comentado anteriormente.
- PSDU (*Physical Service Data Unit*) es la carga útil de la PDU.

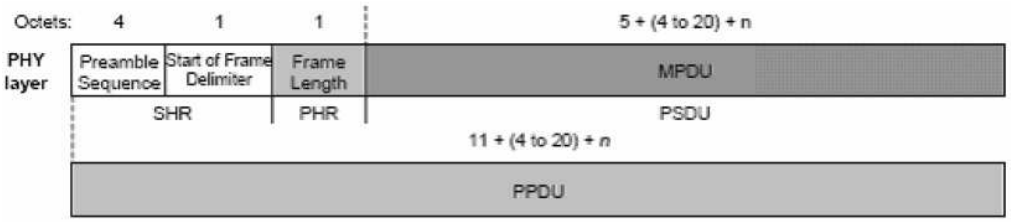

Figura 9 PDU de la capa PHY

2.4. CAPA MAC DEL 802.15.4

La capa MAC proporciona dos servicios: MAC *data service* y MAC *management service*, ambos servicios interactúan en la capa MAC y permiten la transmisión y recepción de tramas, MPDU, a través del servicio de datos PHY.

Las funciones más relevantes de la capa MAC son:

- Generar *beacons* en el caso de ser un *PAN coordinator* y que el resto de nodos se sincronicen al ritmo de los *beacons*
- Mecanismo de acceso al medio CSMA-CA.
- Asociación o desasociación a una PAN.

- Funciones de seguridad (cifrado AES).
- QoS mediante GTS (*Guaranteed Time Slot*).
- Ofrecer un enlace fiable entre dos entidades MAC.

La ventaja de este nivel MAC respecto al de otros estándares es que tan solo se dispone de 21 primitivas de servicio o comandos, lo que redunda en un hardware más sencillo y más barato de fabricar.

2.4.1. FUNCIONAMIENTO

Dentro de la capa MAC del protocolo IEEE 802.15.4 encontramos tres tipos de funcionamiento a la hora de transmitir datos: transmisión de datos utilizando *beacons*, transmisión de datos sin utilizar *beacons* y transmisión de datos utilizando *beacons* con un tiempo de acceso garantizado (GTS). A continuación describiremos los tres tipos existentes:

A. Transmisión de datos utilizando beacons

La transmisión con *beacons* está orientada a redes donde existe el papel del coordinador (topología en estrella), el *PAN coordinator* se encarga de transmitir *beacons* cada cierto tiempo para que los dispositivos dentro de su red se puedan sincronizar. Como se puede observar en la figura siguiente, entre *beacon* y *beacon* se establece una supertrama compuesta por 15 slots (slots de *backoff*) llamados CAP (*Contention Access Period*), mediante los cuales los dispositivos de la red podrán transmitir de forma coordinada.

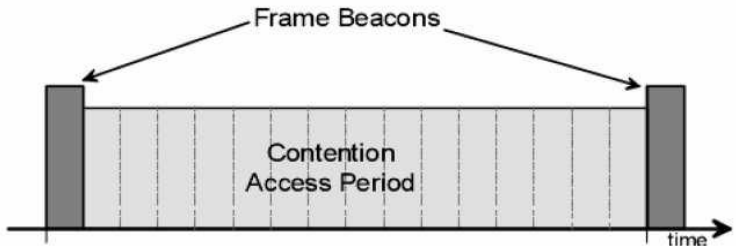

Figura 10 Uso de *beacons*

En este caso, como mecanismo de acceso al medio utilizamos CSMA-CA ranurado en donde las ranuras de *backoff* están alineadas con el comienzo de un *beacon*. Cada vez que un dispositivo desea transmitir, primero tiene que alinearse con el siguiente slot de *backoff* y entonces tiene que esperar un número aleatorio de ranuras de *backoff*. Si el canal está libre, en el siguiente slot comenzaría a transmitir. Si el canal está ocupado, dejara pasar otro número aleatorio de ranuras de *backoff*. Los únicos paquetes que no están sometidos a CSMA-CA son los ACKs y los *beacons*.

B. Transmisión de datos sin utilizar *beacons*

La transmisión con *beacons* está orientada a redes *peer to peer* donde todos se comunican entre todos sin la intervención de un coordinador. El mecanismo de acceso al medio es el CSMA-CA no ranurado en lo que cada dispositivo transmite en el momento que es necesario sin esperar la baliza (*beacon*) de un *PAN coordinator*.

El mecanismo de funcionamiento sería el siguiente: cada vez que un dispositivo desea transmitir datos o comandos MAC tiene que esperarse un tiempo aleatorio, si encuentra el canal libre espera un tiempo de *backoff*, pasado este tiempo intenta transmitir. Si el canal estuviera ocupado después del periodo de *backoff* volvería a esperar otro tiempo aleatorio así sucesivamente hasta llegar a un tope de intentos definidos con la variable BE dentro del algoritmo CSMA-CA.

C. Transmisión de datos utilizando beacons y un tiempo de acceso garantizado

La transmisión con *beacons* y un tiempo de acceso garantizado nos proporciona una latencia determinista para aquellos dispositivos que necesiten tener este parámetro garantizado. Los GTS

vendrán definidos en tiempo en la trama de *beacon* y se sitúan dentro del periodo libre de contienda (*Contention Free Period*), como ilustra la Figura 11. Este espacio está reservado para que en caso de haber mucho tráfico ciertos dispositivos tengan siempre prioridad para lograr mínima latencia.

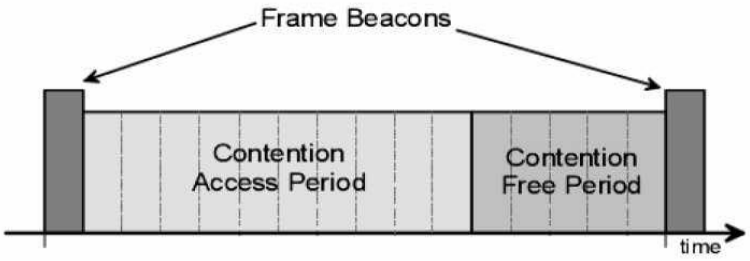

Figura 11 Uso de *beacons* y GTS

2.4.2. MECANISMOS DE ROBUSTEZ

El estándar IEEE 802.15.4 se mueve en entornos hostiles y medios compartidos por lo que se han definido una serie de mecanismos para conseguir que sea más robusto en este tipo de entornos:

- CSMA-CA: Sistema anteriormente comentado basado en la detección de portadora evitando colisiones.
- Paquetes con confirmación (ACK): Cuando enviamos paquetes, se nos devuelve un paquete ACK confirmando que el paquete de datos o cualquier otro ha sido recibido correctamente.
- Verificación de los datos (CRC): Mediante un polinomio generador de grado 16 obtenemos la redundancia y podemos comparar el CRC enviado con el calculado en destino y de esta manera verificar los datos.
- Restricciones de consumo: IEEE 802.15.4 está pensado para aplicaciones que utilicen una batería o una unidad de energía agotable, ya que estas aplicaciones transmitirán información de forma muy esporádica por lo que la cantidad de energía que consume cuando escucha el canal es ultra baja.
- Seguridad: Implementa seguridad de clave simétrica mediante el estándar de encriptación AES, el manejo y gestión de la claves es derivado a capas superiores. Se ha dedicado el capítulo 4 para explicar los mecanismos de seguridad del estándar.

2.4.3. IMPLEMENTACIÓN USADA DURANTE EL PROYECTO

Durante la realización del proyecto, el transceptor radio que se ha empleado ha sido el modelo CC2420 perteneciente a Texas Instruments, antes Chipcon, que implementa el estándar IEEE 802.15.4. Este chipset cumple con todas las características mencionadas en los apartados anteriores, es decir, implementa completamente tanto el nivel físico como el nivel de enlace descritos en el estándar 802.15.4, aunque presenta ciertas particularidades que se detallan a continuación.

Aunque en la capa física el estándar define dos opciones de transmisión según la banda de frecuencia utilizada (868/915 MHz y 2450 MHz), ambos basados en el DSSS, en el modelo de transceptor radio utilizado se trabaja solamente con la banda entre 2400-2483,5 MHz, y por tanto, el tipo de modulación empleado para transmitir es O-QPSK.

Por lo demás, el transceptor radio CC2420 en su capa física presenta las mismas características que las descritas en el estándar 802.15.4, como pueden ser el control del transceptor radio, calidad del enlace (LQI), selección de canal, detector de energía(ED), estimación de la claridad del canal (CCA) para su uso en CSMA-CA a nivel MAC, etcétera

En cuanto a la capa MAC, la implementación de la capa MAC funciona en el modo IEEE 802.15.4 con *beacons* y GTS. Paralelamente al desarrollo de este proyecto se estaba implementando el algoritmo de acceso al medio CSMA/CA, pero no llegó a aplicarse en la toma de medidas.

3. SEGURIDAD DE LA INFORMACIÓN

Las exigencias de seguridad de la información han ido creciendo en las últimas décadas motivadas principalmente por una mayor exposición de los sistemas. Antes del uso extendido de equipos de proceso de datos, la seguridad de la información se garantizaba por medios físicos y administrativos.

En el contexto actual, con entornos distribuidos y descentralizados, se hace indispensable la transmisión constante de grandes volúmenes de datos a través de redes públicas, atravesando multitud de medios. La necesidad de seguridad ha ido creciendo a medida que crecía esta interconexión global, pero también debido al tipo de información en tránsito. De hecho, esto ha provocado la aparición de legislaciones en pos de preservar los derechos de privacidad y de secreto de las comunicaciones, como en el reglamento LOPD en el caso español, la Directiva Europea 95/46/CE en la UE [7] o el Proyecto de Ley Federal [8] en EEUU.

3.1. CONCEPTO DE SEGURIDAD DE LA INFORMACIÓN

La definición clásica del concepto de seguridad de la información viene dada por los requisitos necesarios para preservarla. Para ilustrar estos requisitos, se ha de observar el sistema como una función que transfiere información entre dos agentes. Este flujo está representado en la Figura 12. Según este esquema, los requisitos que caracterizan la seguridad de la información son los siguientes:

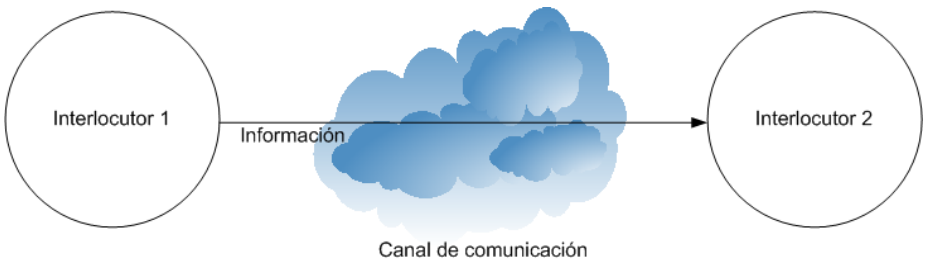

Figura 12 Comunicación básica

3.1.1. CONFIDENCIALIDAD

Exige que la información de un sistema de computadores sea accesible para lectura solamente a aquellas personas o sistemas autorizados. Este tipo de acceso incluye la visualización y otras formas de revelación, incluyendo el simple revelado de la existencia del objeto.

La amenaza a la confidencialidad se encuentra en la intercepción de la comunicación por un agente no autorizado, ilustrado en la Figura 13. La probabilidad de esto ocurra dependerá del medio físico de la comunicación, o de los elementos intermedios ubicados entre los dos extremos de la comunicación.

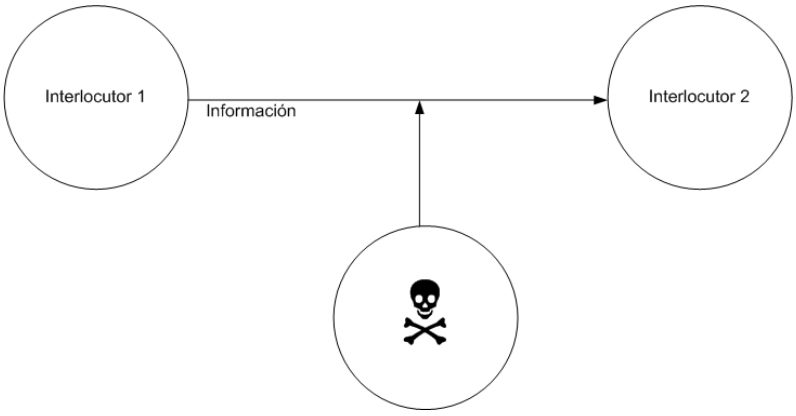

Figura 13 Compromiso de la confidencialidad

3.1.2. INTEGRIDAD

Exige que los elementos de un sistema de computadores puedan ser modificados sólo por aquellas personas o sistemas autorizados. La modificación incluye escritura, cambio, cambio de estado, borrado y creación.

Las amenazas a la integridad vienen dadas por un acceso no autorizado y por la posibilidad de alterar la información en tránsito (Figura 14). Al igual que el caso de la intercepción, la probabilidad de éxito de esta amenaza dependerá de la facilidad del atacante de acceder al canal, pero sus efectos pueden ser muy perjudiciales sin no es detectado.

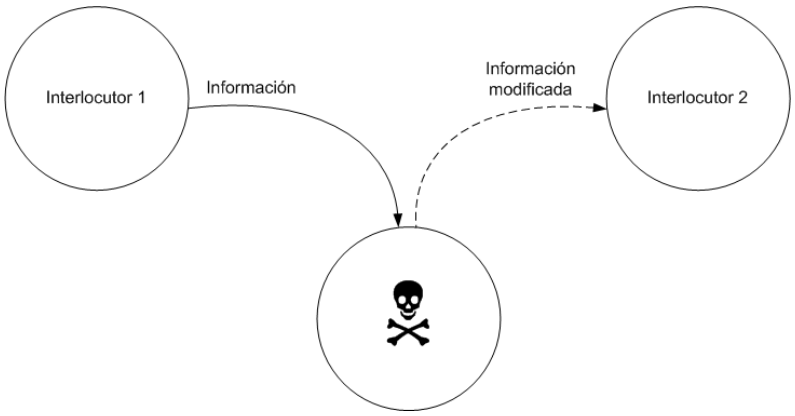

Figura 14 Alteración de la información

Adicionalmente, otro ataque a la integridad consiste en la inserción de información falsa en el sistema, por ejemplo retransmitiendo un paquete, como representa la Figura 15.

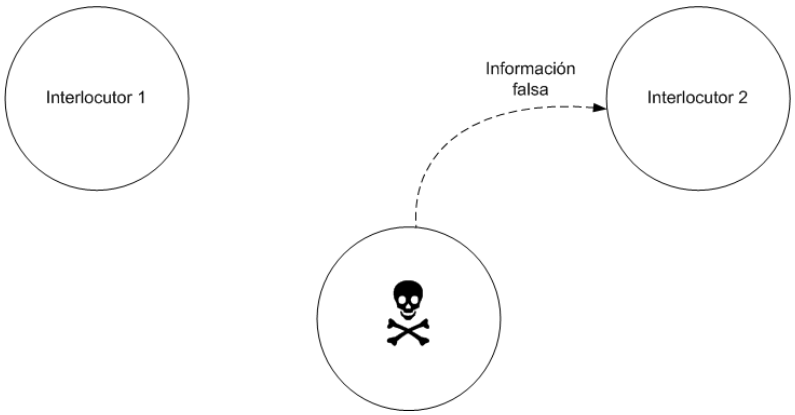

Figura 15 Inyección de información

3.1.3. DISPONIBILIDAD

Exige que todos los elementos de un sistema de computadores estén disponibles a los grupos autorizados.

La amenaza a la disponibilidad se encuentra en la interrupción de las comunicaciones, ya sea interviniendo sobre el medio, sobre los interlocutores o sobre los elementos intermedios involucrados en la comunicación, como ilustra la Figura 16.

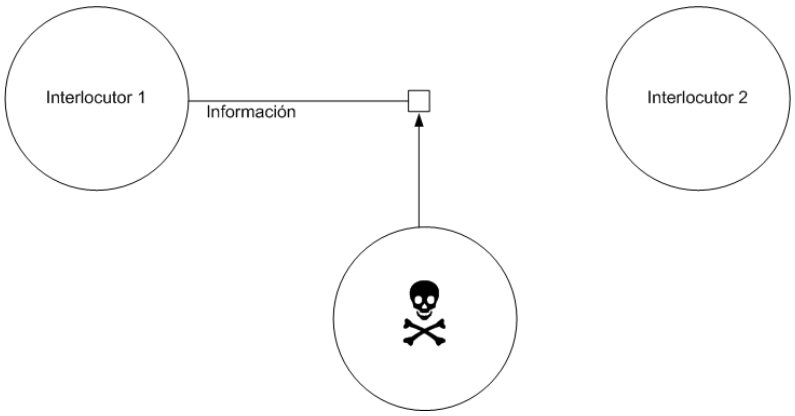

Figura 16 Denegación de servicio

3.2. RIESGOS, AMENAZAS Y VULNERABILIDADES EN REDES DE SENSORES

Observando las amenazas básicas que afectan a un sistema que pretende garantizar la seguridad de la información, en este apartado se particulariza al contexto de las redes inalámbricas de sensores.

La principal característica que va a orientar los ataques a estas redes consiste en la naturaleza del medio de comunicación. Las comunicaciones inalámbricas utilizan el espectro electromagnético, por lo que un atacante con la cobertura adecuada podría interceptar la información sin ser detectado.

Adicionalmente, muchas de las aplicaciones de estas redes se desarrollan en entornos no controlados e incluso hostiles, por lo que la seguridad física de los sensores tampoco puede controlarse.

De estos dos factores se derivan la mayor parte de los riesgos, los cuales afectarán a la información y a la infraestructura. Las medidas de seguridad han de disponer de los mecanismos necesarios para preservar todos estos aspectos:

- La confidencialidad, debido a la facilidad de acceder al canal de comunicación.
- La autenticidad de la información, ya que se transmite por el aire a todos los dispositivos dentro del área de influencia del emisor.
- La integridad de la información transmitida, para evitar modificaciones accidentales o malintencionadas.
- La vigencia de la información, para evitar la retransmisión de información obsoleta.
- La disponibilidad del canal y de los nodos, evitando ataques de denegación de servicio.
- El acceso lógico a la red, el cual debe ser exclusivo a los nodos designados.
- La captura de algún nodo, siendo necesario que el acceso físico al mismo no permita acceder a la información que contiene.
- Evitar la suplantación de los nodos por dispositivos malintencionados, los cuales pueden afectar la integridad mediante la inyección de información falsa o a la disponibilidad de la red, impidiendo el paso de mensajes legítimos o provocando un consumo descontrolado de los recursos de los nodos.

3.3. VECTORES DE ATAQUE

En este apartado se describen una serie de ataques a la infraestructura de las redes de sensores, catalogados según la capa del modelo OSI que se encuentra expuesta. Como se ha comentado, la arquitectura de las redes de sensores puede dividirse conceptualmente en una pila en la que las capas inferiores ofrecen funcionalidades a las capas superiores, abstrayendo los detalles de bajo nivel (Figura 17).

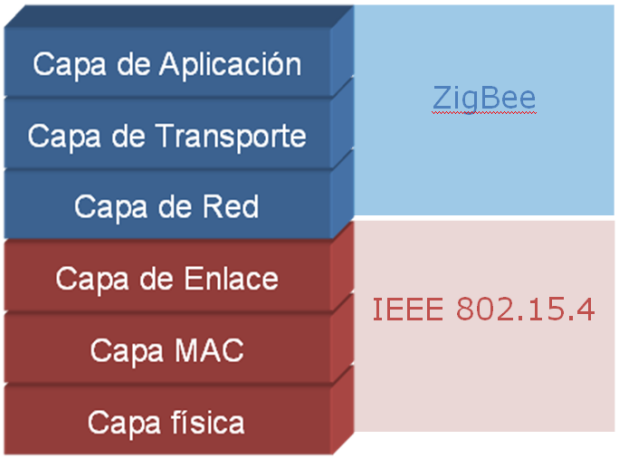

Figura 17 Pila OSI

Teniendo en cuenta el objeto de estudio de este trabajo, la atención debe centrarse en las capas que son responsabilidad del estándar IEEE 802.15.4, ya que las medidas dispuestas en él son las que se han implementado y evaluado. En cualquier caso, se describirán también aquellas que afectan a las capas de ZigBee, ya que la mayoría explotan las vulnerabilidades de las capas inferiores.

Existen dos perfiles diferenciados de ataque, en función de dónde se ubique el atacante con respecto a la red. Si la amenaza no forma parte de la red (*outsider*), verá limitada su capacidad de dañar o intervenir las comunicaciones y deberá utilizar técnicas diferentes frente a nodos maliciosos que hayan conseguido formar parte de la red de sensores (*insiders*). Este último caso es el menos favorable, ya que el daño potencial que puede infligir es mayor, agravando la situación si pasa inadvertido. Por tanto, será prioritario evitar los accesos no autorizados a la red de sensores.

Se ha incluido únicamente una muestra de los ataques más representativos, ya que en la literatura se han descrito multitud de técnicas para explotar las vulnerabilidades de las redes de sensores. Una compilación más exhaustiva de ataques y vulnerabilidades puede encontrarse en [9], [10], [11], [12] y [13].

3.3.1. CAPA FÍSICA

La principal vulnerabilidad en este ámbito se debe a la imposibilidad de asegurar el entorno físico del sensor. Muchas de las aplicaciones de las redes de sensores requieren un despliegue masivo de los mismos sobre entornos desatendidos, por lo que no es posible aplicar medidas preventivas. Si un atacante tiene acceso físico a un nodo, nada le impide destruirlo o capturarlo.

Sin embargo, sí que se cuenta con medidas para evitar el acceso no autorizado a los datos almacenados en el sensor. Por ejemplo, el microcontrolador MSP430 de Texas Instruments cuenta con un fusible que impide el acceso de lectura/escritura a la memoria [14] similar al mecanismo software del Atmel ATmega128 para evitar las funcionalidades de testeo [15].

Otra debilidad inherente a la capa física afecta directamente al medio de transmisión, las interferencias de radio, que pueden ser tanto intencionadas como accidentales. En entornos industriales, la posibilidad de interferencia debido a maquinaria y otros dispositivos es muy elevada, lo que refuerza la idea de implementar un mecanismo que garantice un medio libre de ruidos.

Adicionalmente, la emisión intencionada de interferencias, denominada *jamming*, pueden deshabilitar por completo una red sin necesidad de tener una ubicación próxima a los o disponer de información sobre los protocolos de comunicación que intervienen. Estos tipos de ataques en los que se ocupa el canal de manera continuada suelen ser muy ruidosos y fácilmente detectables, como indican Walters *et al.* en [11].

Por el contrario, el *jamming* reactivo, el cual escucha el canal para provocar colisiones cuando algún sensor emite datos, puede pasar inadvertido, por lo que no se podrán tomar medidas para mitigar el impacto. Existen estudios que plantean una serie de soluciones para evitar este hecho de manera eficiente, como el

descrito en [16], identificando los nodos legítimos que provocan la acción de los *jammers*, y modificando su rol dentro de la red.

Además del efecto inmediato a la disponibilidad del canal de comunicación, estos ataques pueden devenir en que se agoten las baterías de los nodos, debido al intento de retransmisión de las tramas, como se detalla en el siguiente apartado.

El estándar IEEE 802.15.4 establece el uso de DSSS (*Direct Sequence Spread Spectrum*), [5] y [6], en la modulación, la cual presenta resistencia frente al *jamming*, tanto intencionado como accidental, entre otras ventajas. En un nivel superior, hace uso de CSMA/CA y GTS para evitar las colisiones, como se vio en el apartado 2.4, además de una etapa de evaluación del canal antes de transmitir denominado CCA (*Clear Channel Assessment*).

3.3.2. CAPA MAC

Los ataques a nivel MAC requieren mayor conocimiento de la topología y funcionamiento de la red objetivo. Se consideran más sofisticados, aunque en muchos casos requieren un perfil de ataque de *insider*.

Basado en el efecto comentado en el apartado anterior sobre el agotamiento de las baterías, surge el *sleep deprivation torture*, descrito en [12]. Esencialmente, se provocan colisiones intencionadas y continuadas para que se retransmita la información, sin permitir que el nodo pase al estado de espera. Para lograr esto puede utilizarse *jamming* o información legítima, como la inyección de tramas NAK, aunque el receptor haya recibido correctamente los datos.

Utilizando una técnica similar a ésta se encuentran los ataques de repetición, en los que se utilizan tramas legítimas utilizadas anteriormente, como un *beacon*, y se vuelven a transmitir. Esto provoca que la frescura e integridad de la información se vea comprometida. Lógicamente, el estándar IEEE 802.15.4 mantiene un esquema de numeración secuencial para evitar este tipo de situaciones.

En [17], se describe un problema de seguridad derivado de la suplantación de un nodo, denominado *ataque Sybil*. El nodo malicioso presenta numerosas identidades a la red (Figura 18), invalidando la información de los nodos legítimos y modificando la información de rutado. En ese mismo estudio, Quinghua *et al.* proponen un método de detección basado en la protección mutua de los nodos y la contabilidad de las tramas emitidas por cada nodo, comprobando si existen diferencias entre lo contado y lo declarado.

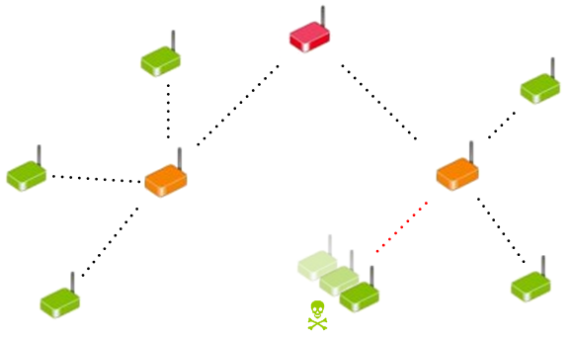

Figura 18 Ataque Sybil

3.3.3. CAPA DE RED

En esta capa, los ataques son llevados a cabo por *insiders* que funcionan como enrutadores (FFD sin ser coordinador), ya que afectan a los algoritmos de rutado de la red. Esencialmente, falsean esta información para provocar la indisponibilidad de la red, entorpeciendo el diagnóstico.

Un ataque que caracteriza este comportamiento es el descrito en [9] y [10], denominado *sinkhole*. Un nodo comprometido presenta rutas de muy buena calidad entre diferentes partes de la red. Esto provoca que toda la información que circule por él, se pierda (Figura 19). Se convierte en un sumidero de información y

son muy difíciles de detectar, sobre todo si se hace una selección de la información que es retenida. En [18] se presenta una solución, pero tienen unos requisitos temporales en la sincronización tan estrictos, que en muchos casos es inviable su aplicación. Para el caso que nos ocupa, es preferible asegurar que no existan nodos no autenticados en la red.

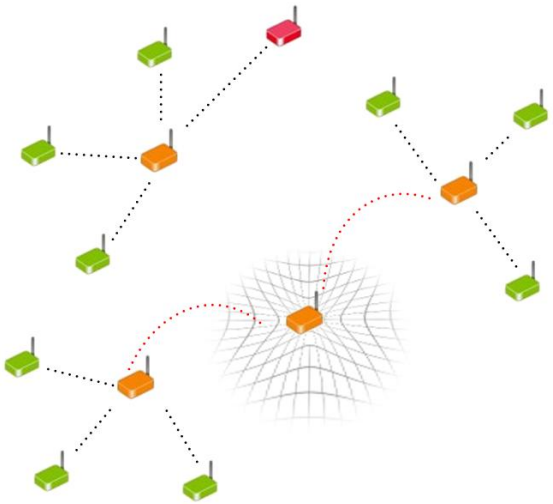

Figura 19 Sinkhole

Por otro lado, en [18] se presenta una variación más sofisticada de este ataque, en la que se utilizan enlaces fuera de banda de baja latencia para falsear la distancia entre los nodos, ya que se tuneliza la información (Figura 20). Esto genera *sinkholes*, difíciles de detectar como se ha comentado, e inmunes a la solución propuesta en el caso anterior. Adicionalmente, los nodos malintencionados no requieren formar parte de la red, ni tener una identidad en la comunicación. Simplemente actuando como *relays* pueden modificar la información de rutado, aún empleando criptografía y autenticación. Frente a este tipo de amenazas se han desarrollado algoritmos que tienen en cuenta la localización geográfica de los *motes* para realizar el rutado [10], aunque quedan fuera del alcance de este estudio.

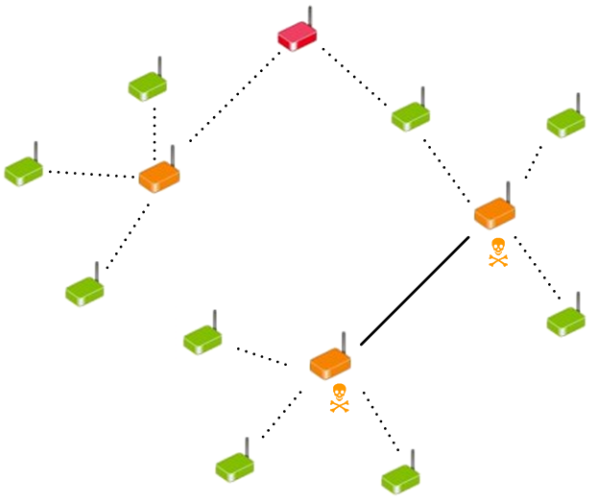

Figura 20 Wormhole

Otro ejemplo de ataque en la capa de red, señalado en [9] y [10], aprovecha una antena de alta ganancia para confundir a la red, presentándose como vecino de un número de sensores, cuando estos no tienen la capacidad de emisión suficiente para comunicarse con la antena. Se denomina *Hello flood* (Figura 21) y provoca el consumo de las baterías, ya que los nodos tratan de responder al anuncio, emitiendo señales al vacío.

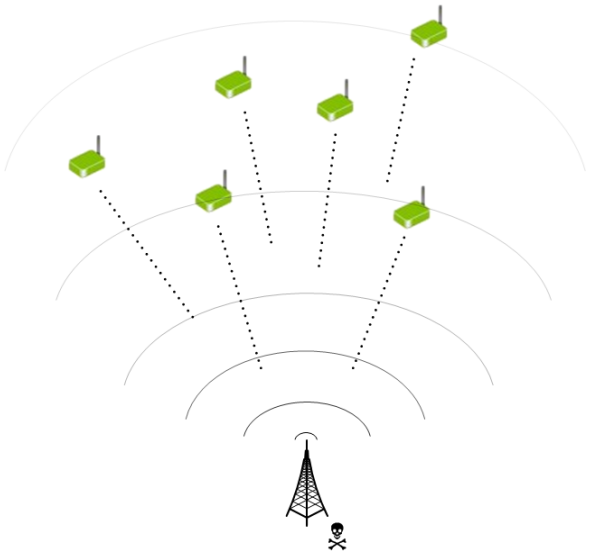

Figura 21 HELLO Flood

3.3.4. CAPA DE APLICACIÓN

En esta capa, las vulnerabilidades asociadas a las aplicaciones dependen de la implementación de las mismas. Así, como en aplicaciones de otras plataformas, es posible explotar los problemas ocasionados por desbordamiento de búferes, *memory leaks* o inyección de parámetros. Estos ataques escapan del ámbito de este estudio por la diversidad de los mismos y porque son fuertemente dependientes de la propia aplicación.

Sin embargo, un problema común será el consumo de recursos asociados al número de conexiones que se establezcan en las aplicaciones, por lo que se deberán provisionar mecanismos para gestionar el número de conexiones permitidas.

4. SEGURIDAD EN IEEE 802.15.4

Como se ha visto en el apartado anterior, el número y naturaleza de las amenazas es muy elevado y puede parecer inabarcable abordar todos estos problemas. Sin embargo, hay que destacar que la mayoría de estos ataques presuponen que se tiene acceso lógico a la red de sensores, como *insider*. Sin este acceso, la explotación práctica de estos ataques es muy complicada, si no imposible, por lo que los esfuerzos han de enfocarse en evitar que un nodo cualquiera pueda asociarse a una red establecida.

Para ello, el estándar IEEE 802.15.4 establece una serie de medidas que permitirán autenticar a los dispositivos que formen parte de la red, inhabilitando la asociación de aquellos que no estén autorizados. En este apartado se describen esas funcionalidades, que son el objeto de estudio de este proyecto.

El estándar establece el algoritmo de cifrado que debe utilizarse en las operaciones criptográficas, sin embargo, no especifica cómo han de gestionarse las claves o las políticas de autenticación que deben aplicarse. Estas tareas deben ser tratadas por las capas superiores, gestionadas por ZigBee.

La información de este apartado se ha obtenido directamente del estándar IEEE 802.15.4, versión 2003 [5], así como la versión de 2006 [6], con el apoyo del trabajo de D. Gascón en [19]. Las diferentes versiones del estándar difieren considerablemente en cuanto a la información necesaria para gestionar la seguridad o, mejor dicho, en cuanto a la estructura de esta información. Se ha seguido la especificación del año 2003, con la excepción de las ACLs, cuya especificación se ha eliminado de la versión de 2006 precisamente porque presentaban vulnerabilidades en caso de utilizar varias entradas ACL con una misma clave [13, 20].

4.1. DESCRIPCIÓN

La seguridad se obtiene del cifrado simétrico, el cual cubrirá los requisitos de confidencialidad e integridad. El algoritmo de cifrado usado es AES (*Advanced Encryption Standard*) con una longitud de claves de 128 bits (16 Bytes). Este algoritmo no sólo se utiliza para cifrar la información, sino también para validarla. Mediante un "código de integridad del mensaje" (MIC), también denominado "código de autenticación del mensaje" (MAC[2]), añadido al final del mensaje, se consigue dotar de integridad a las comunicaciones. Este código asegura la integridad de la cabecera MAC y del *payload*, a la vez que asegura que el emisor es quien dice ser. Se construye cifrando ciertas partes de la cabecera MAC con la clave que establezca la política de gestión de claves, y que será conocida por los nodos que se estén comunicando. Si se recibe una trama de algún nodo no confiable, el código MIC generado no corresponde con el que fue enviado en la trama, al haberse generado con una clave diferente. El MIC puede tener varios tamaños, 32, 64 y 128 bits, aunque siempre se construye utilizando el algoritmo AES de 128 bits. Este tamaño únicamente indica cuántos bits se añadirán al final de cada trama.

La confidencialidad de las comunicaciones se conseguirá cifrando el contenido del *payload* mediante el algoritmo AES y una clave de 128 bits. El funcionamiento de estos modos se detallará en los siguientes apartados.

4.2. FORMATOS DE TRAMA

Para gestionar estas operaciones de seguridad, son necesarios una serie de campos de las tramas IEEE 802.15.4, resaltados en la Figura 22:

- **Frame Control**, ubicado en la cabecera MAC
- **Auxiliary Security Header**, localizado en la cabecera MAC

[2] El acrónimo MAC, relativo al *Message Authentication Code*, puede inducir a la confusión con el de *Medium Access Control*, correspondiente a la capa MAC del modelo OSI. Por este motivo, a lo largo de este documento se referenciará este código como MIC. Aunque MIC sea el acrónimo de *Message Integrity Code*, aporta tanto integridad como autenticación de la manera en que es utilizado en el estándar.

- **Data Payload**, ubicado en el campo de datos MAC

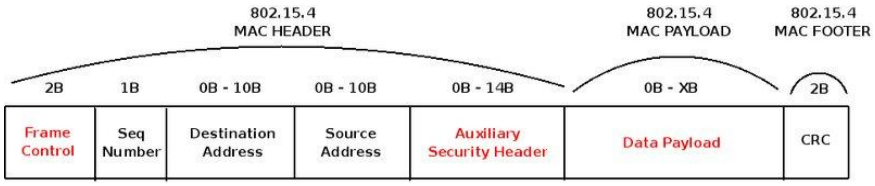

Figura 22 Cabecera IEEE 802.15.4

El *Auxiliary Security Header* solo se active si el bit *Security Enabled* del campo *Frame Control* se encuentra a 1. Esta cabecera especial, representada por la Figura 23, contiene 3 subcampos:

- **Security Control**, indica el nivel de seguridad seleccionado para esta trama.
- **Frame Counter**, es un contador proporcionado por el emisor de la trama para proteger ante ataques de repetición. Por esta razón, cada mensaje tiene un número de secuencia único representado por este campo, no necesariamente correlativo.
- **Key Identifier**, especifica la información necesaria para seleccionar la clave en el nodo receptor. Los nodos han de contener las mismas claves, organizadas de la misma manera.

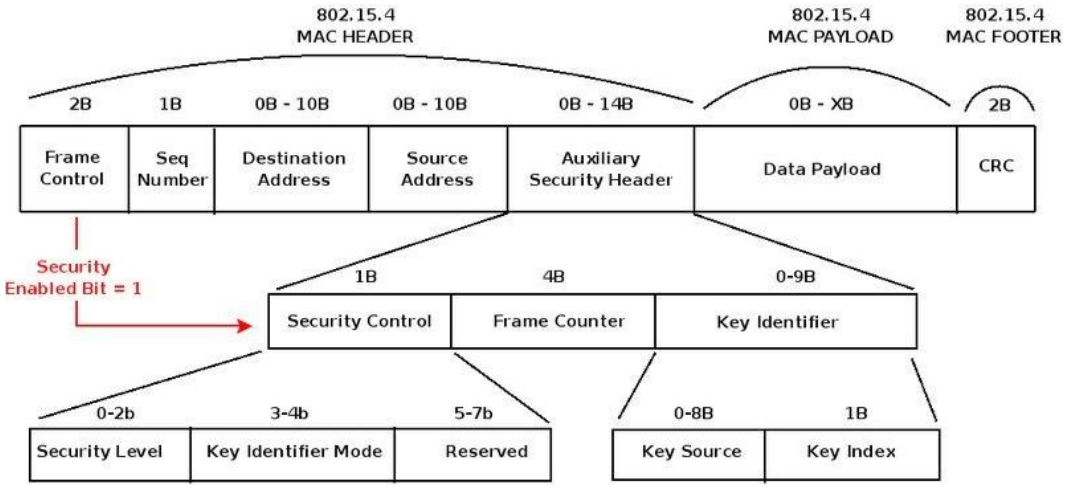

Figura 23 Subcampos de ASH

El subcampo *Security Control* es el lugar donde se ubica la política de seguridad, que seleccionará el modo de funcionamiento de AES, y el modo de identificación de la clave, que puede ser implícito o explícito. El resto del espacio está reservado para posibles ampliaciones.

Los valores posibles de *Key Identifier Mode* son:

- **0**, el valor de la clave es conocido de manera implícita por el emisor y el receptor, por lo que no se especifica en este mensaje.
- **1**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y el parámetro estático *macDefaultKeyStore*.
- **2**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y los 4 bytes de *Key Source*
- **3**, la identificación de la clave se realiza de manera explícita con el byte de *Key Index* y los 8 bytes de *Key Source.*

Según esta configuración, el número máximo de claves que pueden utilizarse es de 2^{72} claves, lo que implicaría un consumo máximo de memoria para las claves de 2^{72} x 16B, lo que no es factible en ningún sistema. Lo importante de este aspecto no es el número en sí, si no que el número es suficientemente grande para que sea escalable a diferentes políticas de gestión de claves.

La implementación que se hace de la seguridad indica que ésta se efectúa a razón de cada trama. Esto quiere decir que un receptor, para cada trama recibida, tendrá que seleccionar la clave correspondiente, actualizar los valores de contador y realizar la operación criptográfica correspondiente. Teóricamente, un mismo nodo emisor podrá enviar dos tramas diferentes a un mismo receptor con niveles de seguridad distintos, lo que aporta una gran flexibilidad. Por ejemplo, se podrán enviar tramas *beacon* garantizando la identidad del coordinador, pero sin cifrar el contenido, y utilizar un cifrado completo en caso de enviar tramas de datos.

4.3. NIVELES DE SEGURIDAD

Los niveles de seguridad que ofrece IEEE 802.15.4 se especifican en el subcampo *Security Level* de la cabecera auxiliar de seguridad. Estos niveles definen el modo de funcionamiento del algoritmo AES, proporcionando autenticación, confidencialidad o ambas.

Los 3 bits de este campo permiten seleccionar entre 7 niveles de seguridad, desde lo más bajo, que no realiza ninguna operación criptográfica, hasta el nivel que ofrece más garantías. La siguiente tabla especifica las características de cada nivel.

Valor	Suite de cifrado	Operación
0	Sin seguridad	Datos en claro. Autenticación sin validar.
1	AES-CBC-MAC-32	Datos en claro. Autenticación validada.
2	AES-CBC-MAC-64	Datos en claro. Autenticación validada.
3	AES-CBC-MAC-128	Datos en claro. Autenticación validada.
4	AES-CTR	Datos cifrados. Autenticación sin validar.
5	AES-CCM-32	Datos cifrados. Autenticación validada.
6	AES-CCM-64	Datos cifrados. Autenticación validada.
7	AES-CCM-128	Datos cifrados. Autenticación validada.

Tabla 3 Niveles de seguridad

Esencialmente, lo que indica esta tabla es que existen 3 modos diferenciados (CBC-MAC, CTR y CCM) de realizar las operaciones criptográficas, y que aportarán funcionalidades diferentes. En los siguientes apartados se detalla el funcionamiento y particularidades de cada modo de funcionamiento.

4.3.1. ADVANCED ENCRYPTION STANDARD

AES, también conocido como Rijndael, es un estándar de cifrado adoptado por el Gobierno de EEUU, tras 5 años de estudios para sustituir al vulnerable DES [21]. Es uno de los algoritmos más populares de cifrado simétrico y existe multitud de literatura sobre su seguridad frente al criptoanálisis y a ataques prácticos[3]. De hecho, la política de la NSA respecto a su uso, indica que puede utilizarse para la información clasificada como TOP SECRET [22].

La criptografía simétrica requiere que los interlocutores compartan la clave de cifrado (de ahí el "simétrico") y ésta debe ser segura. La fortaleza del cifrado depende exclusivamente de la clave, por lo que, a mayor tamaño de clave, mayor seguridad. En [23], el NIST establece una guía de buenas prácticas y recomendaciones para garantizar que las claves cumplan los siguientes requisitos:

[3] Existe cierta polémica en la comunidad criptográfica sobre la seguridad de AES. Se han obtenido resultados que reducen el espacio de búsqueda de claves de 128 bits a sólo 2^{100} posibilidades. Académicamente se puede considerar el algoritmo como "roto", aunque en la práctica, el espacio de búsqueda sigue siendo un problema computacional inabordable.

- Han de ser generadas de manera aleatoria para reducir la probabilidad de que un atacante las deduzca o sean reutilizadas
- Han de cambiar frecuentemente para reducir la posibilidad de descubrimiento mediante criptoanálisis
- Han de ser protegidas en almacenamiento, para que comunicaciones anteriores no puedan ser descifradas
- Han de ser protegidas durante su transmisión
- Deben ser completamente eliminadas cuando no sean necesarias

Dado que, para que dos pares se comuniquen de manera privada es necesario que ambos compartan una clave, la criptografía simétrica no escala bien en ese tipo de comunicaciones. Se necesitarían *n(n-1)/2* claves para comunicar *n* nodos entre sí de manera privada, consumiendo 16 bytes por cada clave. Con 37 nodos se superaría la cantidad de RAM disponible en el microcontrolador MSP430 (10KB).

AES funciona bajo un esquema de cifrado por bloques, lo que significa que los mensajes que se han de cifrar son separados en porciones de tamaño fijo, los bloques. Estos bloques sufren una transformación invariante para obtener los datos cifrados. En esencia, el cifrador AES es una caja negra que tiene dos entradas, un bloque de datos de tamaño fijo y una clave del mismo tamaño. Con estos dos componentes genera un bloque da datos cifrados, también del mismo tamaño. Si el numero de bloques de un mensaje es mayor que 1, entonces se deberá utilizar uno de los modos de operación disponibles.

Estos modos garantizarán diversos grados de confidencialidad o integridad. Como se ha visto, IEEE 802.15.4 especifica 3 de esos modos:

- CBC-MAC para la autenticación (e integridad)
- CTR para la confidencialidad
- CCM para la confidencialidad y la autenticación (e integridad)

En el caso que el mensaje no sea múltiplo de 128 bits, el último bloque deberá rellenarse antes de entrar al cifrador. Cómo ha de rellenarse no se especifica, pero existen una serie de convenciones para ello. Existen ataques que explotan esta característica del cifrado por bloques. Sin embargo, dependerá del modo de operación. Así, CTR no sufre de este problema, ya que realiza una XOR con el resultado del cifrador, desechando los bits sobrantes.

4.3.2. CBC-MAC

El primero de estos modos se utilizará para autenticar los mensajes, como indica el acrónimo MAC. Cada bloque toma como entradas el resultado de su anterior y el bloque de mensaje correspondiente para generar un bloque del mismo tamaño con la información cifrada, con excepción del primer bloque, que se inicializa a 0 (Figura 24).

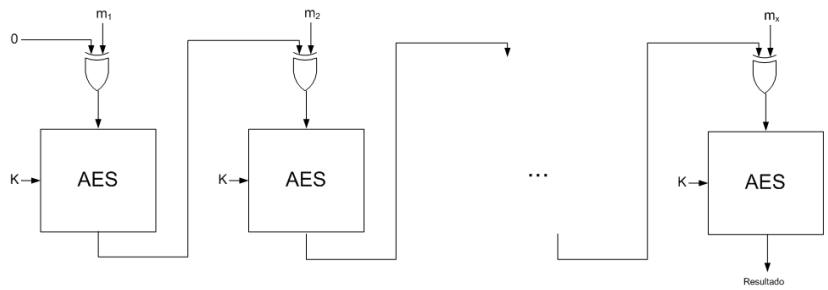

Figura 24 AES-CBC-MAC

De esta manera, al final se obtendrán 128 bits generados a partir del mensaje y la clave, formando un resumen o *hash* criptográfico del mensaje. Este resultado se anexa al mensaje que se pretende enviar. Cuando el receptor quiere comprobar la validez del mensaje, solo tiene que realizar el mismo cálculo y compararlo

con el resumen anexado. Si coincide, significa que quien ha enviado el mensaje conoce la clave de cifrado y que el mensaje no ha sido modificado en tránsito. De esta manera se obtiene autenticación e integridad.

Es posible parametrizar el tamaño del mensaje de autenticación (el resumen generado). De los 128 bits que se generan, se pueden anexar los 128 bits completamente, o bien únicamente los 64 o 32 bits menos significativos. De ahí las tres versiones que especifica el estándar IEEE 802.15.4. Cuanto mayor sea el tamaño del MIC, mayor será el espacio de búsqueda para un ataque de fuerza bruta. Concretamente, un atacante debería probar 2^{32} códigos MIC para autenticar un mensaje mediante CBC-MAC32. Este número puede llegar a ser asumible por un atacante si se dispone del tiempo suficiente y la clave permanece constante. Afortunadamente, el algoritmo no permite deducir la clave de cifrado a partir del código MIC falseado. Aún así, es recomendable que las claves se renueven periódicamente.

4.3.3. CTR

El modo CTR, llamado así porque hace uso de un contador como vector de inicialización, se utiliza para cifrar el contenido de los mensajes, aportando confidencialidad. A cada bloque del mensaje se le aplica una función XOR con la salida del cifrador, el cual ha generado un valor de 128 bits a partir de la clave y un vector de inicialización (Figura 25).

Ese vector de inicialización, de 128 bits, está formado por un *nonce* (*number used once*) y un contador de bloque, que se irá incrementando en función del bloque que tenga que cifrar. El *nonce* debe construirse con información conocida por el emisor y el receptor, y suele estar contenida en la cabecera de la trama a enviar. El motivo de esta configuración es para evitar que mensajes idénticos resulten en mensajes cifrados idénticos, ya que esos mensajes serían susceptibles al criptoanálisis y a la rotura del cifrado. Los bloques se cifran de manera independiente, por lo que podría aprovecharse el paralelismo para cifrar todos los bloques simultáneamente.

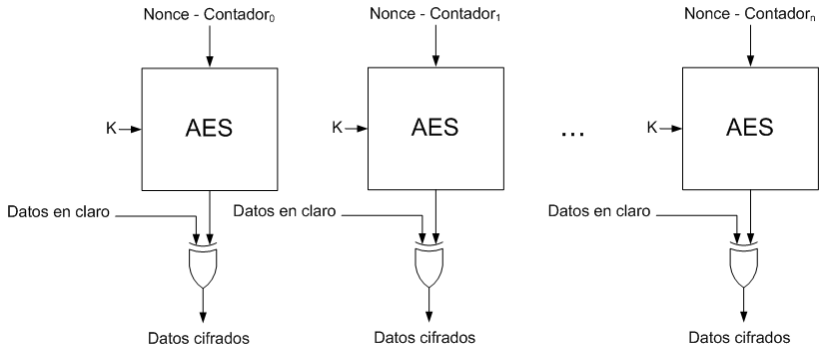

Figura 25 AES-CTR

Para descifrar un mensaje, se ha de seguir exactamente el mismo procedimiento, ya que si $x \oplus y = z$, entonces se cumple $x = y \oplus z$. Así, al realizar una operación XOR entre la salida del módulo de cifrado (cuyo vector de inicialización deberá ser igual al del emisor) y los datos cifrados, el resultado serán los datos en claro.

Es necesario destacar que existen estudios que desaconsejan el uso de CTR, e incluso proponen su eliminación de la especificación del estándar [13]. Esto es debido a que, al no contar con un control de la integridad del mensaje, un atacante podría modificar el contenido cifrado del *payload* y del CRC sin ser detectado, abriendo un vector que podría llegar a afectar a la confidencialidad. La explotación de esta vulnerabilidad depende del protocolo específico de la aplicación, por lo que no se puede hacer una afirmación que englobe a todos los despliegues, pero sí que es cierto que, a la larga, es una puerta abierta a incidentes de seguridad.

4.3.4. CCM

El modo CCM, descrito en el RFC 3610 [24], combina los dos modos anteriores en uno solo, aportando confidencialidad, autenticación e integridad. El coste de esto es que ha de realizar dos pasadas sobre el mensaje; la primera de ellas para generar el MIC, y una segunda para cifrar el *payload* y el MIC (Figura 26). La única diferencia es que en el cálculo del MIC, el vector de inicialización se corresponde con el que se utilizará en el cifrado.

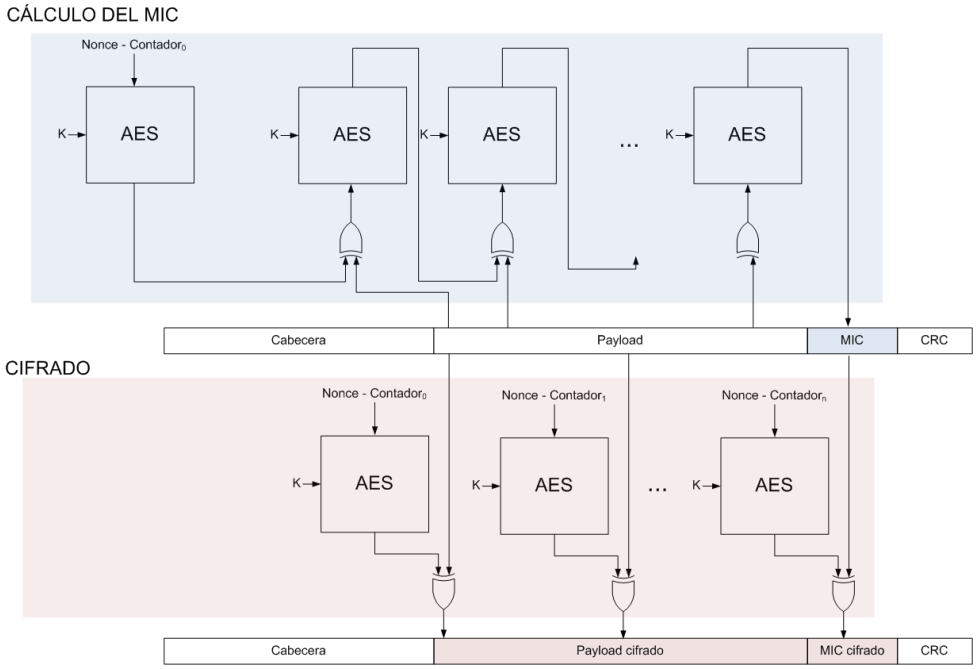

Figura 26 AES-CCM

5. ENTORNO DE TRABAJO

En este capítulo se van a presentar y describir los componentes, tanto *hardware* como *software*, que han sido necesarios para llevar a cabo este análisis.

El kit de desarrollo del que se ha dispuesto para la realización del trabajo está compuesto de:

- 2 Placas básicas Shimmer
- 1 Programador/cargador USB Shimmer
- 1 Sniffer Radio ZENA Wireless Network Analyzer
- 1 Osciloscopio digital DSO-2100, para la toma de medidas

5.1. SHIMMER

En las figuras adjuntas, obtenidas de [25] al igual que el resto de la información, se muestran dos fotografías de la placa básica de Shimmer y sus componentes. El elemento esencial de la plataforma es el microcontrolador de bajo consumo MSP430 de Texas Instruments, que se comunica con el resto de periféricos a través de los módulos de expansión.

En los siguientes apartados se describen los módulos más importantes de la plataforma Shimmer que se han empleado en el proyecto.

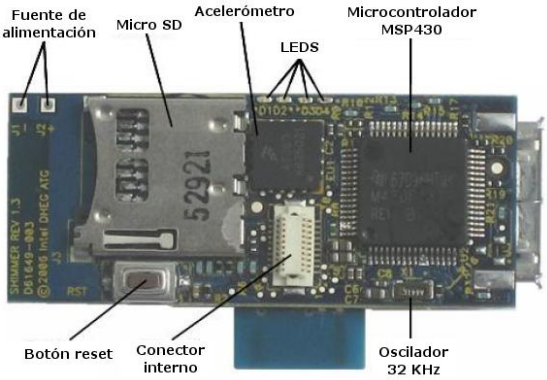

Figura 27 Anverso del Shimmer

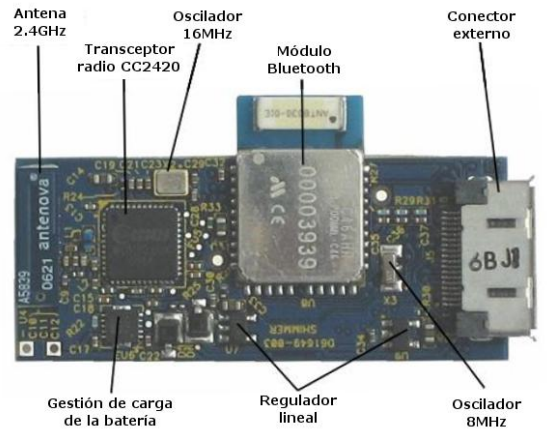

Figura 28 Reverso del Shimmer

5.1.1. MPS430

Como se ha comentado, el elemento esencial de las placas Shimmer es el microcontrolador MSP430. Su principal ventaja reside en su bajo consumo de energía durante los períodos de inactividad. Los 10 KB de

RAM disponibles es el máximo disponible en la familia de estos procesadores y ofrece capacidades de *buffering* mejoradas. La arquitectura del procesador es RISC de 16 bits, con dos buses de memoria. Dispone de 10KB de memoria RAM, 40 KB de memoria Flash, donde se ubicará el SO y la aplicación, y 128 bytes para información interna.

Para comunicarse con el chip de radio 802.15.4, utiliza una conexión serie síncrona por USART en modo SPI, utilizando un reloj maestro compartido entre los dos componentes.

Cuenta con ocho canales para conversiones analógico/digital, que se utilizarán en la adquisición de los datos de los módulos sensores. Tres de ellos se utilizan en el acelerómetro incluido en la placa, otros tres se utilizan en el conector interno de expansión y los dos restantes en el conector externo, que se ha utilizado únicamente para programar las placas y alimentarlas. Como el resto de componentes, pueden desactivarse cuando no están en uso para reducir el consumo de energía.

Dispone también de 4 LEDs para utilizarlos a discreción del programador. Durante la implementación se han utilizado como indicadores del estado del dispositivo, pero se desactivaron durante la toma de datos para no "ensuciar" las medidas.

5.1.2. RADIO

La placa cuenta con dos módulos de radio, Bluetooth y 802.15.4, pudiendo funcionar de manera dual. Para el modo 802.15.4, utilizado en el desarrollo de este proyecto, la placa monta el transceptor CC2420 de Chipcon (ahora Texas Instruments). Este módulo está diseñado para aplicaciones de bajo consumo y baja velocidad, y permite desconectarlo desde el MSP430 para reducir el consumo.

La radio modula directamente en DSSS y presenta una tasa teórica de transferencia de 250Kbps. Ofrece soporte para aplicaciones como el manejo de paquetes, transmisión de datos, potencia de la señal recibida, calidad del enlace, temporización de paquetes y obviamente cifrado, con lo que reduce el trabajo del MSP430 y del programador.

5.2. HURRAY Y TINYOS

La implementación usada en este proyecto está basada en el trabajo previo realizado por Cunha *et al.* en [26]. Este trabajo se realizó sobre el sistema operativo basado en eventos TinyOS, diseñado específicamente para las redes de sensores y sobre placas MICAz y TELOSB. Debido a las particularidades de TinyOS, desde un principio se optó por portar este código a FreeRTOS, un sistema operativo en tiempo real.

Gran parte del tiempo dedicado a este proyecto se invirtió en comprender el funcionamiento de TinyOS y la implementación Hurray del estándar. En este apartado se describen las principales características de estos dos componentes *software*.

5.2.1. TINYOS Y NESC

TinyOs es un sistema operativo de código abierto diseñado específicamente para redes de sensores capaz de ejecutar código a alta velocidad. Al ser un sistema operativo basado en eventos es muy adecuado para trabajar con sensores ya que la actividad de un nodo sensor está basada en los impulsos externos que recibe: mensajes radio de otros nodos sensores, detección de sonido, detección de una temperatura más alta de lo habitual, etc.

Debido a las limitaciones de los dispositivos que ejecutarán TinyOS, se establecen una serie de restricciones que no existen en otros sistemas:

- Las zonas de memoria se localizan en memoria estática.

- No existe memoria dinámica. No hay punteros.
- Una única pila asignada la tarea que se ejecuta en un momento dado.
- No existe protección de memoria.

TinyOS utiliza un modelo de programación basado en el concepto de "wiring" (enlazar o cablear) componentes software para producir un programa final. Este modelo de programación además, pone requisitos sobre cómo deben ser escritos los programas.

Los componentes de TinyOS pueden dibujarse formando una pirámide invertida: encima de todo se encuentran los componentes del nivel de aplicación, mientras que la capa más baja la ocupan los componentes que se asientan directamente sobre el hardware. En este sentido, TinyOS presenta tres abstracciones de computación: los eventos, los comandos y las tareas:

- Los comandos representan "llamadas hacia abajo": un componente llama a comandos que pertenecen a componentes que se encuentran por debajo de él para solicitar un servicio.

- Los eventos representan "llamadas hacia arriba": un componente advierte que ha sucedido un evento a componentes que se encuentran por encima de él. Se implementan mediante capturadores de interrupción, por lo que deben ser rápidos para no paralizar la CPU.

- Las tareas son un mecanismo para computación asíncrona de larga duración. Una tarea se ejecuta sincronizadamente respecto a otras tareas, todas tienen la misma prioridad y se ejecutan en orden. Sin embargo, una tarea puede verse obligada a abandonar el procesador si llega un evento de mayor prioridad. Por lo tanto si realizamos tareas con requisitos de tiempo real, éstas deben ser cortas para evitar que puedan ser interrumpidas por sucesivos eventos.

Por otro lado, NesC (*Network embedded systems C*) es un lenguaje de programación, de sintaxis similar a C, en que está programado el sistema operativo TinyOS, así como sus librerías y aplicaciones.

Originalmente, TinyOS estaba programado en C (hasta su versión 0.6), sin embargo, en la transición entre la versión 0.6 y la 1.0 se reimplementó todo el sistema operativo en NesC. La principal ventaja que proporciona NesC es que dispone de manera nativa de las abstracciones necesarias que se utilizan en TinyOS, añadiendo funcionalidades, como la detección de errores de "wiring".

Un programa en NesC estará estructurado mediante componentes, y el programador construirá las aplicaciones conectando componentes ya creados con otros nuevos. De esta manera, toda aplicación estará dividida lógicamente en tres partes: Configuración, Interfaces y Módulos.

La especificación de los componentes se realiza mediante los interfaces y representan el único punto de acceso a un componente de manera bidireccional. Una interfaz declara un conjunto de comandos que el proveedor de la interfaz debe implementar, y otro conjunto eventos que han de ser implementadas por el usuario de la interfaz. Un mismo componente puede provisionar o utilizar varios interfaces y múltiples instancias de un mismo interfaz.

Los módulos proporcionan el código de la aplicación, implementando una o varias interfaces, mientras que las configuraciones se encargan de ensamblar los componentes. Para ello, definen las asociaciones entre los interfaces utilizadas por unos, con las interfaces proporcionadas por otros.

Finalmente, la ejecución de una aplicación sobre TinyOS tendrá dos hilos de ejecución, el determinado por las tareas y el determinado por los capturadores de eventos, que lanzarán a ejecución una serie de instrucciones como respuesta a un evento asíncrono externo.

5.2.2. HURRAY

La implementación de partida que se utilizó en las fases preliminares del proyecto estaba basada en el proyecto Hurray del Instituto Politécnico de Oporto [26]. Se basa en la especificación del estándar IEEE 802.15.4 del 2003, y la plataforma objetivo son los motes MICAz y TELOSB, en lugar de Shimmer.

Las funcionalidades del estándar que proporciona la versión utilizada (v1.2) son las siguientes:

- Arbitraje de acceso al canal mediante el algoritmo CSMA/CA
- Mecanismo GTS para garantizar el uso del canal a ciertos nodos
- Gestión de *beacons*
- Soporte a la construcción de tramas, pero únicamente con el formato de direccionamiento corto
- Mecanismo de asociación y desasociación
- Gestión de la MAC PIB, la base de información de la red
- Escaneos del canal con detección de energía y mediante la detección de *beacons*

Las partes del estándar que no se han incluido incluyen subcampos de las tramas, la versión no ranurada de CSMA/CA, y la seguridad. Por tanto, la implementación de este proyecto no se ha basado en código de terceros y se ha construido desde cero.

Como se ha comentado, Hurray se utilizó como base en las fases iniciales del proyecto, pero se desechó este trabajo por la baja calidad del código en algunos puntos. De esta manera, el departamento desarrolló una capa MAC simplificada, contando con las funcionalidades básicas, a las que se le añadió las características de seguridad.

5.3. FREERTOS

El modelo de componentes usado en TinyOS resulta complicado de gestionar en aplicaciones de cierto tamaño, con la dificultad añadida del lenguaje NesC, por lo que se optó por portar la implementación MAC a un sistema operativo en tiempo real para sistemas empotrados.

El paradigma de FreeRTOS se ajusta perfectamente a los requisitos temporales de la capa MAC 802.15.4 y a las limitaciones hardware de las placas Shimmer. Los sistemas operativos en tiempo real garantizan los resultados de manera determinista en un tiempo determinado, y FreeRTOS está diseñado para sistemas empotrados, por lo que también cumple con las restricciones relativas a los recursos del sistema.

Los mecanismos que ofrece FreeRTOS para construir aplicaciones son las tareas, el planificador, los semáforos, la colas y las rutinas de atención a interrupciones.

Las tareas son el bloque básico de los programas, y no son más que funciones de C que se repiten indefinidamente. Están asociadas a una prioridad, para que el planificador pueda determinar qué tareas han de ejecutarse en primer lugar. El planificador es configurable, y puede funcionar con expropiación o de manera cooperativa. Para gestionar la temporización de las tareas existen un par de funciones que permiten suspender las tareas durante un tiempo determinado. De esta manera, es posible definir tareas períodicas (como el envío de *beacons*) o introducir retardos en espera de eventos externos. Hay que tener en cuenta que los retardos pueden variar si existen tareas de mayor prioridad en ejecución.

Para comunicar las tareas se utilizan las colas y son gestionadas por el SO. Se utilizan por tanto llamadas al sistema para enviar y recibir datos, con la ventaja que el sistema puede bloquear esas llamadas cuando la cola se encuentre llena o vacía, respectivamente. Estas estructuras se han utilizado en la generación de los eventos de las capas PHY y MAC, como la recepción de tramas. Concretamente, la tarea quedará suspendida a la espera de que llegue un dato a la cola, durante un tiempo definido por un *timeout.* Para evitar problemas de sincronización cuando las tareas efectúen operaciones sobre recursos compartidos, se dispone de semáforos para arbitrar el acceso. Estos mecanismos de sincronización se utilizan en el acceso al búfer de tramas listas para enviar.

Las rutinas de atención a interrupción permiten ejecutar funciones de manera asíncrona debido a señales del *hardware*. De esta manera, es posible activar una serie de procedimiento cuando llega una nueva

trama a la radio. Sin embargo, hay que tener en cuenta que estas funciones han de ser rápidas, ya que interrumpen en flujo normal del programa, bloqueando la tarea que estuviese en ejecución en ese momento. Es imprescindible que la rutina de interrupción no se bloquee, ya que provocaría una latencia intolerable y poco predecible

Adicionalmente, no se deben invocar funciones que puedan conmutar tareas, a no ser que explícitamente se indique que la función fue invocada por una rutina de interrupción. Existe el riesgo de que el planificador conmute a una tarea de mayor prioridad, condenando a la rutina de interrupción a esperar a que terminen las tareas de mayor prioridad. FreeRTOS proporciona métodos para evitar que se den estos casos.

6. FUNCIONALIDADES IMPLEMENTADAS

En este apartado se describen las operaciones y funcionalidades que se han implementado en la capa MAC de los sensores Shimmer. Se ha seguido la especificación del estándar IEEE 802.15.4 del 2003, teniendo en cuenta las limitaciones de la implementación sobre FreeRTOS. Estas limitaciones no afectan a las funcionalidades ni su comportamiento, sino a la gestión dinámica de los parámetros, que deben definirse en tiempo de compilación. Por lo demás, el comportamiento es exactamente el mismo que en un sistema en producción.

Las fuentes de referencia para poder llevar a cabo esta implementación han sido las dos versiones del estándar del IEEE, [5, 6], así como los manuales de desarrollo de la plataforma Shimmer, tanto para el chip de radio CC2420 [27], como del microcontrolador MSP430 [14].

Como se ha comentado anteriormente, el código de partida estaba formado por un nodo coordinador, responsable del envío periódico de *beacons* y el mantenimiento del modo GTS, y por un nodo sensor, encargado de recopilar las adquisiciones de un sensor ECG, empaquetarlas en una trama del tamaño máximo permitido y enviarlas al nodo coordinador en su *slot* de tiempo.

La implementación realizada para este proyecto ha añadido las capacidades de cifrado expuestas en el estándar, tratando de preservar la modularidad del código. Para ello, se definió una capa adicional, *MAC Security*, que cuenta con la mayor parte de las operaciones de seguridad. Las responsabilidades de esta capa se resumen en estos puntos:

- Configuración del *hardware* de radio para habilitar las operaciones criptográficas
- Inicialización de los parámetros necesarios, tanto en el *hardware* de radio como en la capa *software*.
- Actualización de los parámetros de seguridad.

La capa encargada de lanzar los comandos adecuados al hardware es la capa MAC, y es necesario que sea así, por lo que esa funcionalidad se implementó en el código de la capa MAC, preservando las funcionalidades anteriores.

Por otro lado, ha sido necesario modificar las funciones responsables de la construcción de las tramas, ya que la activación de la seguridad implica cambios en la estructura de las mismas, así como la previsión necesaria para reservar espacio de una parte del *payload* para almacenar el MIC, sólo en los casos necesarios.

En resumen, se han añadido las capacidades criptográficas sin modificar el código existente, y añadiendo funciones adicionales que no estaban implementadas en el código base.

6.1. SOPORTE HARDWARE

Los niveles de seguridad definidos en el apartado 4.3 han sido implementados mediante el soporte *harware* que ofrece el chip de radio CC2420. Este dispositivo ofrece un módulo criptográfico basado en AES de 128 bits, así como una implementación interna de los modos de operación del estándar.

Todas las operaciones criptográficas están basadas en el cifrado AES con claves de 128 bits. Estas operaciones se realizan dentro de los búferes de transmisión y recepción, a razón de una por trama. Adicionalmente, se incluye un modo de cifrado *standalone*, en el que se cifra el contenido de un búfer de 128 bits directamente, sin vectores de inicialización. Este modo se ha implementado para realizar pruebas de funcionamiento, pero no se le ha dado aplicación práctica.

Existen estudios que confirman la intuición que una implementación *hardware* será más eficiente en términos de tiempo que una implementación *software*. Healy *et al.* realizan en [28] una implementación *software* del algoritmo AES, comparando los datos obtenidos con los resultados del método *standalone* del CC2420, concluyendo que el uso de *hardware* dedicado reduce significativamente el coste de securizar los datos. Sin embargo, hay que destacar que el tiempo necesario para inicializar los parámetros del *harware* tiene un coste significativo, debido al uso del interfaz USART para comunicarse con la RAM del transceptor de radio. En [28] se incluye una tabla que estima que el tiempo de configuración se encuentra un orden de magnitud por encima que el tiempo necesario para cifrar de un bloque de 128 bits. Estos resultados se complementan con los realizados por Hansen en [29].

6.1.1. CLAVES

Para cualquier operación criptográfica es necesario especificar una clave de 128 bits. La memoria RAM del CC2420 reserva dos posiciones de 128 bits para dos claves independientes. Para cualquiera de las operaciones de cifrado o descifrado es necesario seleccionar una de estas claves en memoria. El estándar IEEE 802.15.4 no especifica cómo realizar la gestión de las claves, es responsabilidad de las capas superiores. Un requisito del estándar es que cada clave se utilice sólo para un modo de operación [5], para evitar la posibilidad de criptoanálisis. Esto significa que habrá que mantener una serie de claves en la memoria RAM del microcontrolador, y cargarlas en la memoria de la radio cuando sea necesario.

Las políticas de gestión de claves más comunes que son apropiadas para las redes de sensores según [13], son las siguientes:

- **Clave de red compartida**: Una misma clave es compartida por todos los dispositivos de la red. Los costes asociados a esta política en términos de recursos son mínimos, pero si la clave de un nodo es comprometida, toda la red se ve afectada.
- **Clave por pares**: Se utiliza una clave individual para cada nodo. El compromiso de un nodo únicamente afecta a ese nodo. Como se detalló en el apartado 4.3.1, el coste en memoria puede ser prohibitivo.
- **Claves de grupo**: Se definen grupos de sensores que utilizarán una única cklave de cifrado, por lo que todas las comunicaciones entre dos nodos cualesquiera de ese grupo utilizarán esa clave. Para comunicaciones entre dos nodos de distintos grupos, se utilizarán claves adicionales, tantas como grupos se hayan definido. Presenta problemas por la reutilización de *nonces*, por lo que no es una alternativa segura.
- **Políticas híbridas**: Se utiliza una política de pares en los enlaces con la estación base o coordinador, mientras que el resto de enlaces se cifran con una clave de red compartida.

En este caso se ha optado por una clave de red compartida, empleando dos claves estáticas, definidas en tiempo de compilación e instaladas manualmente en todos los nodos. Este es uno de los problemas del cifrado simétrico, siendo el establecimiento, intercambio o renovación de claves la parte más comprometida de todo el proceso de cifrado. Existen métodos seguros de intercambio de claves sobre canales inseguros, como Diffie-Hellman [30], pero consumen bastantes recursos computacionales. Existen alternativas, como la criptografía de curva elíptica [31], que ofrece las mismas prestaciones con un consumo más moderado de los recursos.

6.1.2. VECTORES DE INICIALIZACIÓN

Los modos de seguridad especificados necesitan unos valores de inicialización, denominados *nonces* (o contador, en caso de CTR). Estos valores iniciales han de ser únicos para cada uso de la clave, para evitar que los envíos sucesivos de una trama resulten siempre en la misma trama cifrada. Adicionalmente, estos vectores requieren ser reconstruidos por los nodos receptores, para inicializar correctamente el módulo criptográfico y descifrar correctamente los mensajes.

1 byte	8 bytes	4 bytes	1 byte	2 bytes
Flags	Dirección de origen	Contador de trama	Secuencia de clave	Contador de bloque

Figura 29 Formato de IV

Para ello, los vectores se definen a partir de 5 campos conocidos:

- **Flags:** Se utilizan para distinguir los modos CTR y CCM. En CCM especifica la longitud del MIC. Se obtiene del nivel de seguridad.
- **Dirección de origen:** Dirección larga del dispositivo emisor. Presente en la trama.

- **Contador de trama:** Representa el contador de la trama saliente del dispositivo emisor. Presente en la trama.
- **Secuencia de clave:** Contador de la clave utilizada, establecido por las capas superiores. Presente en la trama.
- **Contador de bloque:** Contador interno del módulo criptográfico. Se actualiza automáticamente en el *hardware*.

6.1.3. MODOS DE OPERACIÓN

El hardware de la radio permite operar de dos maneras, *standalone*, cifrando mensajes de 128 bits directamente, o *in-lin*e, en la que se realizan las operaciones sobre los búferes de envío y recepción.

El modo *standalone* se ha implementado con propósito de pruebas, pero no se le ha dado ningún uso práctico. El funcionamiento es bastante simple. Se almacenan los datos en una región concreta de la memoria de la radio, se selecciona la clave de cifrado de las dos disponibles y se lanza el comando SAES. Una vez completado el cifrado, se almacena en la misma dirección de memoria donde estaba, sobrescribiendo los datos en claro.

El modo *in-line*, por el contrario, utiliza los búferes de envío y recepción como entradas del módulo criptográfico. Las operaciones se harán en base al valor de longitud de la trama, para no afectar al búfer completo, por lo que es necesario determinar correctamente el tamaño de la trama. Esto es especialmente importante en los modos CBC-MAC y CCM, ya que añaden información a las tramas.

Antes de poder utilizar este modo, es necesario definir las claves, los *nonces* (excepto en CBC-MAC) y los registros de control. Estos registros se configuran en función del nivel de seguridad seleccionado en la capa MAC y son los responsables de configurar el *hardware*.

Cuando la seguridad está activada, las operaciones *in-line* se inician de dos maneras, lanzando el comando STXENC, que cifrará el búfer pero no lo enviará, o mediante los comandos STXON y STXONCCA, que cifrarán y harán la transmisión inmediatamente. Lo que implica esto es que se pueden añadir las operaciones de seguridad sin modificar las funciones de la capa PHY que invocan al hardware de la radio, ya que el comando utilizado es STXON.

En el caso de la recepción, el comando necesario es SRXDEC, que descifrará el contenido del búfer de recepción. En este caso es preciso modificar las llamadas a la capa PHY, ya que antes de descifrar es necesario saber si hay que hacerlo, por lo que se debe leer del búfer parte de la cabecera, concretamente el campo *Frame Control*. Es importante no leer datos de las partes que haya que descifrar, porque el puntero de lectura no puede retroceder. Adicionalmente, el *hardware* dispone de medidas para evitar la lectura mientras se ejecutan las operaciones de descifrado.

6.2. NIVELES DE SEGURIDAD

La definición de los niveles de seguridad se ha realizado de manera estática, al igual que su selección. Esto permite que los nodos se comuniquen utilizando la misma suite de seguridad, o que uno de los nodos no utilice la seguridad, pero no permite el uso de suites diferentes. Este comportamiento no supondría un esfuerzo en el desarrollo si la infraestructura de construcción de tramas fuera dinámica.

6.2.1. CBC-MAC

El principal problema a solucionar es el de añadir información de seguridad a los mensajes, por lo que hay que reservar una cantidad de bytes al final de cada trama. Para ello, se ha caracterizado el incremento de espacio que supone cada modo de operación, descontando una cantidad de bytes al tamaño máximo de la trama. Así, CBC-MAC32 reserva 4 bytes para el MIC, CBC-MAC64 reserva 8 bytes, y CBC-MAC128 reserva 16. También hay que tener en cuenta este tamaño para modificar el campo *Frame Lenght*.

Otro parámetro importante de la implementación de este modo de operación, consiste en especificar el *offset* que hay que dejar al principio de las tramas para calcular la autenticación. En este caso se ha utilizado la trama completa, incluyendo la cabecera MAC, por lo que el desplazamiento es 0. Obviamente, en los modos CTR y CCM no podrá ser así, porque se cifraría toda la trama.

Cuando se activa este modo en la transmisión, el MIC generado se almacena en el buffer, previamente a su envío. En el caso de la recepción, se genera el MIC y se compara con el alojado en el búfer de recepción. Si ambos coinciden, la autenticación estará garantizada y el último byte del MIC es reemplazado por 0x00. En caso de no coincidir, el valor almacenado será 0xFF.

La capa MAC se hace responsable de confirmar que la trama recibida es válida. En caso de no serlo, desecharía el paquete o enviaría una notificación a la capa superior. En este proyecto se ha optado por mostrar un mensaje de depuración, para comprobar que la comunicación se realizaba correctamente.

6.2.2. CTR

En este modo, ha de especificarse el primer byte desde el cual se va a cifrar/descifrar el contenido. De acuerdo al estándar IEEE 802.15.4, solo el *payload* MAC debe ser cifrado, por lo que es preciso calcular la posición del primer byte del *payload*, en función de la cabecera que se haya definido.

Es necesario realizar pasos adicionales previos al cifrado, para ajustar el valor del vector de inicialización, en este caso denominado contador. Como se ha descrito anteriormente, es fundamental conocer la dirección de origen y el contador de trama, disponibles localmente. Tanto este valor como el del desplazamiento del cifrado serán actualizados en el controlador de la radio. En el momento inmediatamente anterior al cifrado, se ha de realizar una espera activa para comprobar que el módulo criptográfico se encuentra libre.

Cuando se inicia el proceso de cifrado, al ejecutar el comando STXON, el contenido del búfer de transmisión se cifra, siempre que haya datos en él. Si no hay datos, esperará la llegada de éstos, cifrándolos a medida que se escriben en el búfer. En esta implementación, la trama completa se encuentra ubicada en el búfer antes del activar el cifrado.

Para llevar a cabo el descifrado, se leerán los primeros bytes de la trama para determinar la suite de cifrado y se calculará la posición del primer byte del *payload*. Adicionalmente, se reconstruirá el valor del contador con los valores del *Auxiliary Security Header*, para inicializar correctamente el módulo criptográfico.

Una vez que todos los valores necesarios se encuentran en la RAM del CC2420, se procederá al descifrado de la trama con el comando SRXDEC, y se esperará a que termine el proceso antes de leer el búfer de recepción. Una vez leído, la trama descifrada es tratada como una trama MAC, pasándola a una capa superior o realizando otra operación, dependiendo del tipo de trama.

6.2.3. CCM

Este modo combina los dos anteriores, teniendo en cuenta que es necesario definir la posición del primer byte del *payload*, lo que implica que se realizará el cálculo del MIC sobre estos datos, a diferencia de CBC-MAC, que lo realizaba sobre la trama completa. Tras la primera pasada del módulo criptográfico para calcular el MIC, se realizará el cifrado sobre el *payload* y el propio MIC, sobrescribiéndolos. Teniendo en cuenta el orden de operaciones, para descifrar será necesario descifrar la trama en primer término, y luego recalcular el MIC para compararlo con lo recibido. Estos pasos los realiza automáticamente el *hardware* de la radio.

6.2.4. STANDALONE

El cifrado *standalone* permite cifrar bloques de 128 bits de manera independiente. Para ello se utiliza una dirección de la memoria RAM del CC2420, se selecciona una clave de cifrado y se lanza el comando SAES a la radio. En este modo no existe inicialización, al tratarse de bloques únicos de 128 bits.

Este modo se ha implementado, pero no se ha hecho uso de su funcionalidad en las pruebas. Puede resultar de utilidad en momentos puntuales, como la creación de números pseudoaleatorios, aunque habría que estudiar sus implicaciones de seguridad si éstos se difundiesen.

6.3. PROCEDIMIENTO DE MEDIDA

El objetivo de este estudio consiste en determinar el impacto del uso de la seguridad especificada en el estándar IEEE 802.15.4 en el consumo de energía. Una vez implementadas las operaciones criptográficas mediante el *hardware* dedicado del dispositivo, se define una batería de pruebas para poder cuantificar el consumo durante el uso del módulo criptográfico.

6.3.1. DESCRIPCIÓN

El uso del módulo criptográfico supondrá un gasto extra de energía a las operaciones de transmisión de datos mediante IEEE 802.15.4. El objetivo es cuantificar ese gasto y determinar qué parámetros son los que influyen en el incremento del consumo. Para cuantificarlo, es necesario determinar el nivel de intensidad de corriente que circula por el circuito cuando se activa la seguridad.

Mediante un osciloscopio digital se han tomado las medidas de voltaje y tiempo durante la transmisión de tramas, utilizando como referencia los valores obtenidos cuando el nodo transmite la información en claro. Las variaciones de voltaje permiten determinar los cambios en la intensidad provocados por la transmisión de las tramas y, junto a las variaciones en el tiempo de transmisión, permiten determinar la energía consumida. De este modo se cuantificarán las diferencias entre los distintos modos de seguridad con respecto a la operativa con ausencia de medidas de seguridad.

El circuito utilizado, detallado en la Figura 30 y compuesto por el nodo sensor y una resistencia en serie, emplea una resistencia de 13Ω para calcular la intensidad del circuito. La sonda 1 permite determinar la diferencia de potencial existente en el Shimmer, que se mantuvo relativamente constante durante todas las pruebas. La sonda 2, por su parte, determina las diferencias de potencial debidas al uso de la radio y del módulo criptográfico. Las gráficas que se presentan al final del capítulo corresponden a las lecturas de esta sonda.

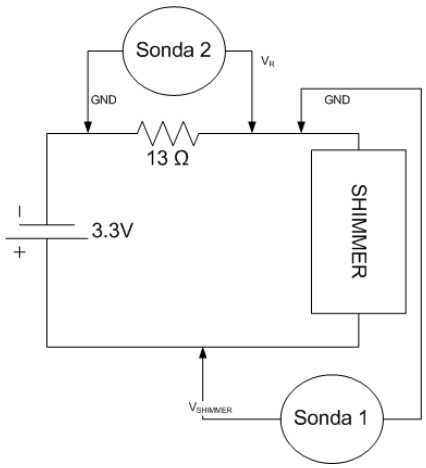

Figura 30 Circuito de pruebas

6.3.2. FUNCIONAMIENTO DEL NODO COORDINADOR

El nodo coordinador únicamente se encarga de transmitir *beacons* periódicamente, sin cifrar, para no provocar sobrecarga debido al descifrado. También es responsable de recibir las tramas del nodo sensor y

determinar si la seguridad se ha aplicado correctamente, descrifrando o autenticando las tramas. Este nodo estaba conectado a un interfaz serie, volcando los datos recibidos y verificando la corrección de las tramas recibidas.

Durante todas las pruebas y mediciones se ha utilizado el mismo nodo coordinador, sin modificar ninguno de sus parámetros.

6.3.3. FUNCIONAMIENTO DEL NODO SENSOR

El nodo emisor se encarga de la adquisición de medidas mediante el sensor incorporado, muestreando el conversor A/D según una tasa de muestreo definida externamente, para establecer el tamaño exacto de las tramas de datos.

Este nodo es el que realiza las labores de cifrado y autenticación de las tramas, justo en el momento previo a la transmisión RF, por lo que es el que está sujeto a las mediciones.

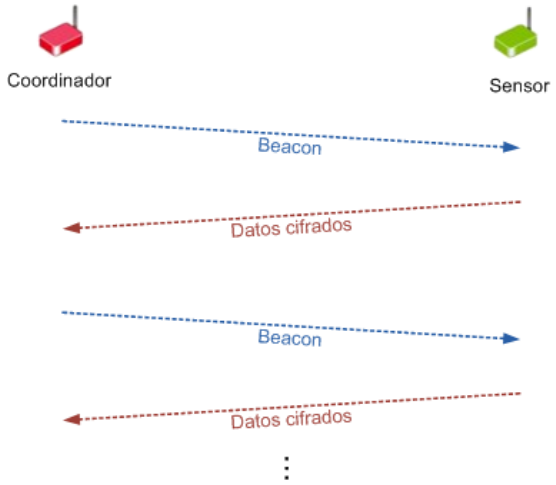

Figura 31 Diseño de las pruebas

6.4. PRUEBAS REALIZADAS Y RESULTADOS

Se han realizado pruebas de todos los niveles de seguridad especificados en el estándar IEEE 802.15.4, parametrizando en cada uno de ellos el tamaño de las tramas. Los tamaños de trama utilizados han sido 12, 24, 48 y 96 bytes, aunque el tamaño máximo de trama es de 128 bytes.

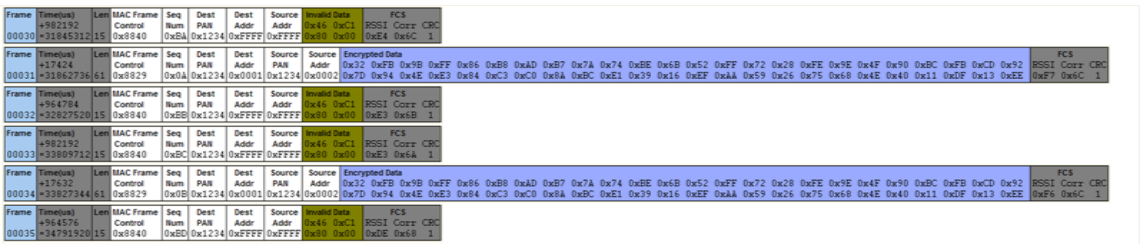

Figura 32 Captura de tramas

Esto es debido a la sobrecarga de espacio necesaria en los modos de autenticación. Teniendo en cuenta que el tamaño máximo es de 128 bytes, que el modo de autenticación que más espacio consume requiere 16 bytes, que la cabecera MAC varía hasta un tamaño máximo de 11 bytes en estos ejemplos y los 2 bytes del CRC, restan 99 bytes para la carga útil. Tomando múltiplos de 12 bytes, obtenemos 96 bytes como el valor máximo de *payload* en estas pruebas.

En el Apéndice se incluyen todas medidas realizadas, así como un resumen de los datos extraídos, como energía consumida, tiempo de transmisión y *overhead* debido al tamaño del MIC.

Las lecturas obtenidas de la sonda 2, representadas en la Figura 33, muestran un pico inicial, debido a la recepción del *beacon*, un tiempo de espera con apenas actividad (la CPU se apaga y sólo se despierta en cada *tick* del sistema operativo), y un pico debido a la transmisión de la trama.

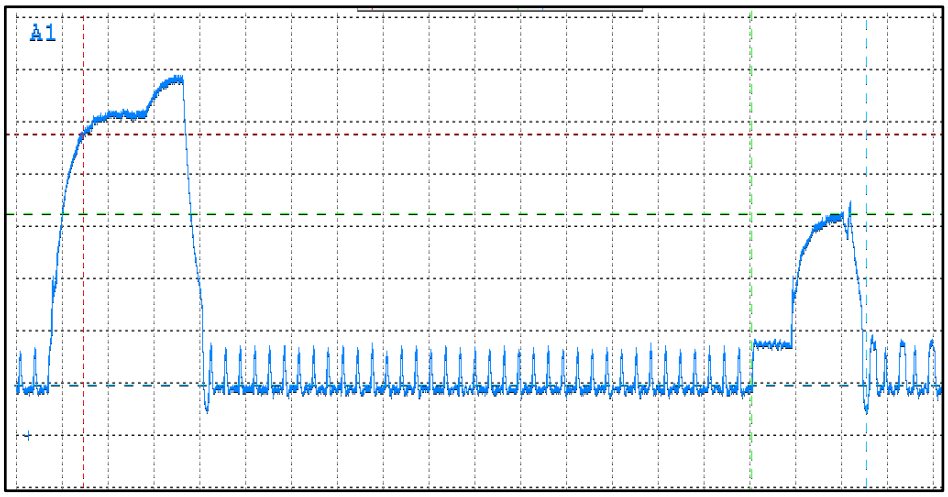

Figura 33 Lecturas de tensión V_R sin seguridad

La muestra anterior se realizó con el nodo emitiendo en el nivel de seguridad 0, esto es, sin ninguna medida criptográfica. La forma de la onda indica actividad de consumo moderado al inicio, debido a la transmisión de la trama por la interfaz serie USART a la RAM de la radio, donde se ubica en búfer de transmisión. El siguiente pico de tensión se produce en la transmisión por radio de la trama.

Si lo comparamos con la forma de onda que resulta de la misma trama aplicando el nivel de seguridad CTR, Figura 34, las diferencias no son apreciables a simple vista. La activación del módulo criptográfico se realiza tras la transmisión serie y antes del envío por radio, pero no se aprecia ningún pico característico en esa franja. Para comparar correctamente los valores resultantes, se ha optado por tomar el valor medio de tensión desde que se inicia la transmisión de la trama hacia la memoria de la radio hasta que termina su retransmisión por radio.

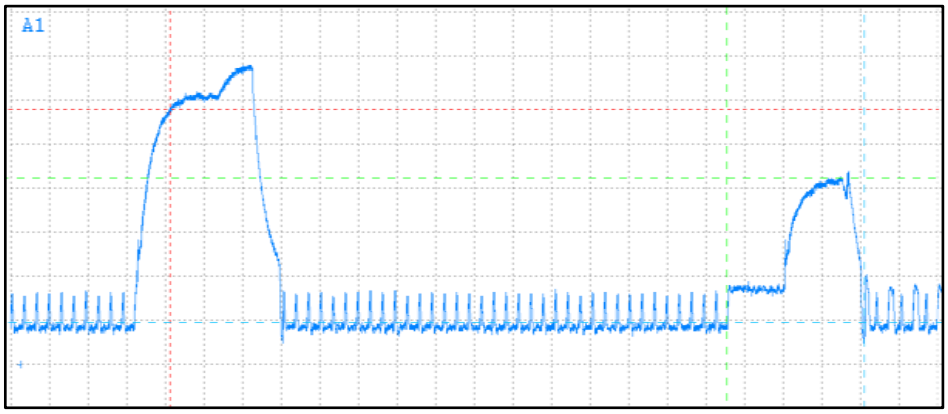

Figura 34 Lecturas de tensión V_R con cifrado (CTR)

Tras la toma de datos de las pruebas, se han organizado los resultados de la siguiente manera:

- **Análisis de la energía consumida**: Se compara el consumo medio de energía en todos los niveles, en función del tamaño de la trama. De esta manera es posible observar la tendencia de cada nivel de seguridad según el tamaño de la trama.
- **Análisis del tiempo de transmisión**: Se comparan los tiempos de transmisión en todos los niveles, en función del tamaño de la trama. Este análisis resulta interesante, ya que determina la causa del incremento en el consumo de energía.

- **Análisis del espacio útil de trama**: Se comparan los incrementos de tamaño de las tramas debido a la seguridad en todos los niveles. Este análisis puede hacerse sin realizar ningún tipo de lectura o implementación, ya que se deduce de la propia especificación del estándar IEEE 802.15.4.
- **Cuadrante del consumo frente al tiempo de transmisión**: En este análisis se pretenden mostrar los niveles de seguridad con respecto al coste en consumo y a tiempo de transmisión.

6.4.1. ANÁLISIS DE CONSUMO

La siguiente figura muestra el consumo absoluto de energía durante la transmisión en los diferentes niveles de seguridad, agrupados por el tamaño de la trama. El valor de referencia, NO_SECURITY, permanece en todos los casos por debajo del resto de niveles de seguridad, aunque no se aprecia el impacto de cada uno de los niveles.

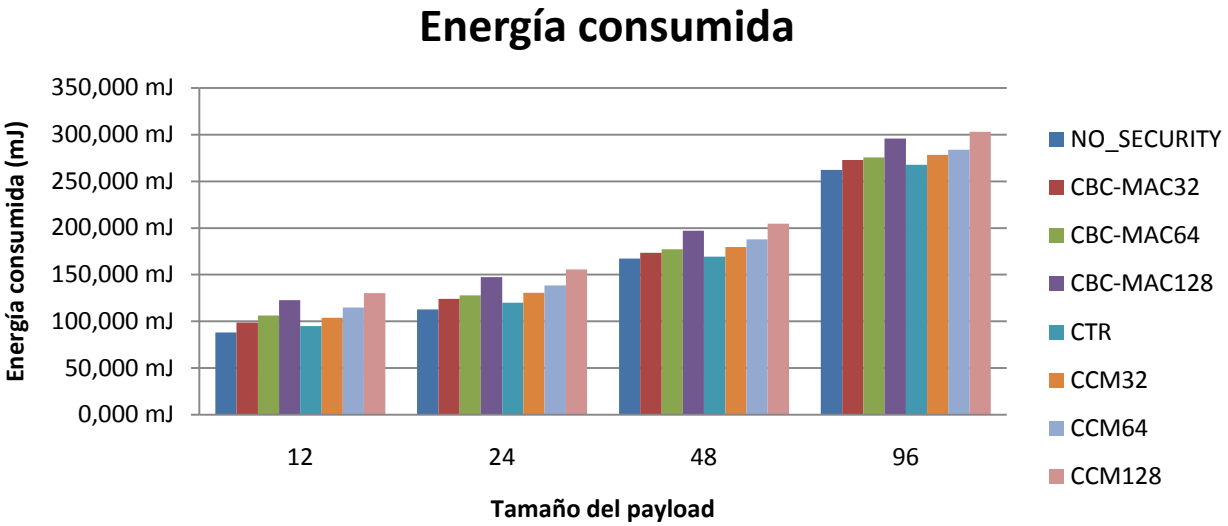

Figura 35 Consumo de energía

Para determinar el impacto, se han normalizado los resultados con respecto a la referencia NO_SECURITY y se presentan como el porcentaje de energía consumida debido a la habilitación de la seguridad. La Figura 36 ilustra el comportamiento de los niveles de seguridad a medida que se aumenta el tamaño de trama. Por ejemplo, se observa que la habilitación del modo CTR incrementa entre un 7,5% y 2,5% la energía consumida durante la transmisión. Resulta interesante comprobar cómo, a medida que se aumenta el tamaño de la trama, el gasto debido a seguridad disminuye.

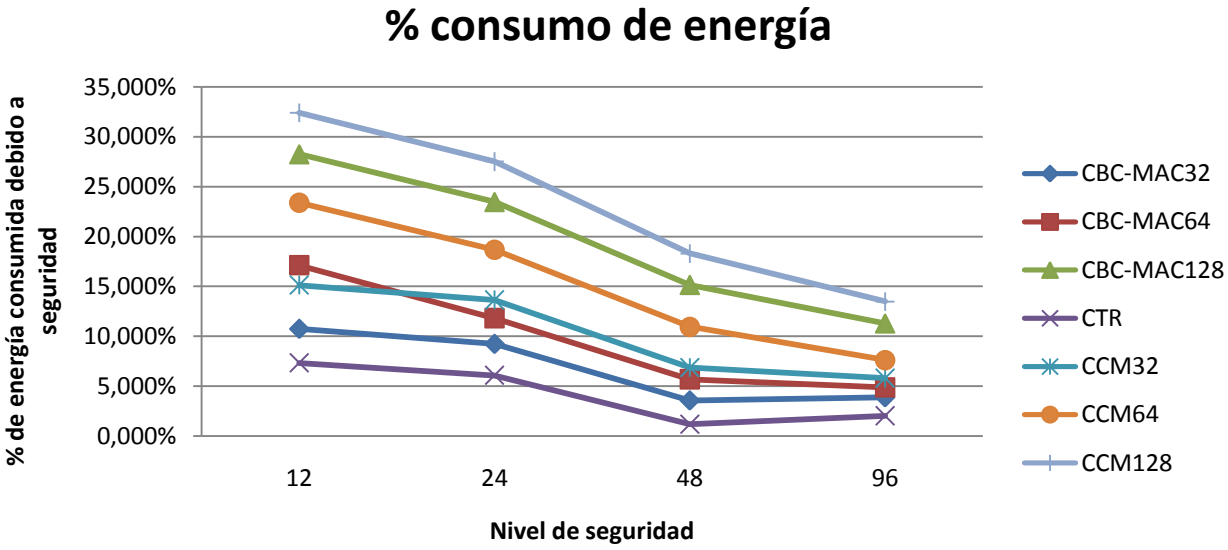

Figura 36 Impacto en el consumo

Observando cuidadosamente los datos relativos al nivel de seguridad CTR, el cual no provoca ningún incremento en el tamaño de trama, se aprecia que el impacto que supone su uso es muy poco significativo, tanto en términos de consumo como de tiempo de transmisión. Sin embargo, muestra una tendencia similar al resto de modos de seguridad, disminuyendo su impacto a medida que el tamaño de la trama aumenta, cuando debería mantenerse más o menos constante, independientemente del tamaño.

La explicación a este fenómeno reside en el tiempo necesario de configuración del *hardware* previo al cifrado. Como se ha comentado en este capítulo, es necesario cargar la clave en la memoria de la radio y actualizar el contador por cada trama enviada (escribiendo también en memoria). Este tiempo es significativamente mayor que el tiempo que requiere el cifrado de un bloque de 16 bytes. De esta manera, el tiempo de *setup* (que se puede considerar independiente del tamaño de trama a cifrar) supone mayor coste cuando el número de bloques de cifrado es pequeño.

6.4.2. ANÁLISIS DEL TIEMPO

En cuanto a los tiempos de transmisión, la Figura 37 muestra el consumo absoluto de tiempo durante la transmisión. Obviamente, a mayor tamaño de trama, mayor será el tiempo de transmisión. Los niveles CBC-MAC128 y CCM128 presentan los peores resultados, ya que añaden 16 bytes de datos en el MIC que han de ser transmitidos.

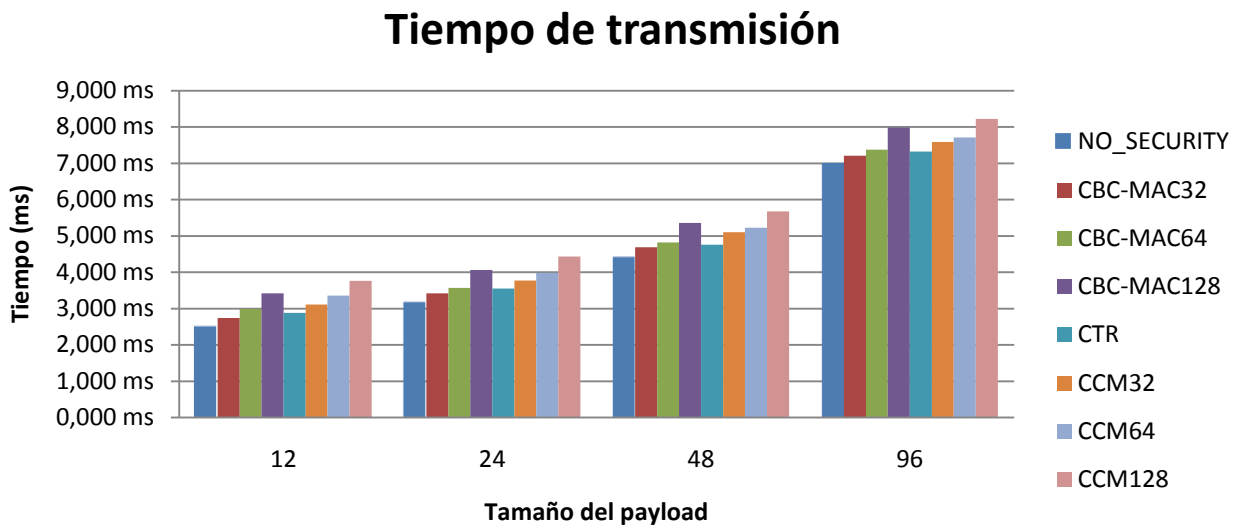

Figura 37 Tiempos de transmisión

Continuando con el razonamiento del apartado anterior, para evaluar el impacto de la seguridad, es preferible normalizar los resultados. En la Figura 38 se muestra el porcentaje del tiempo de transmisión debido a la habilitación de seguridad. Se observa una tendencia lineal descendente a medida que aumenta el tamaño de la trama y el *overhead* debido al MIC supone cada vez menos respecto a la trama.

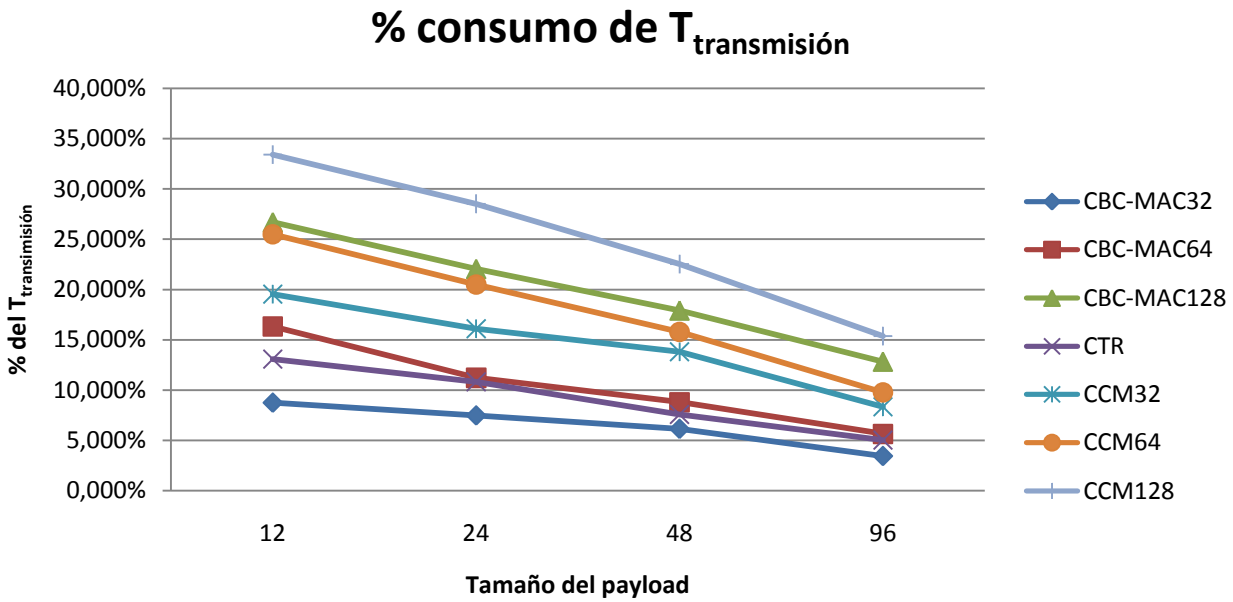

Figura 38 Impacto en el tiempo

6.4.3. ANÁLISIS DEL TAMAÑO ÚTIL

Este último análisis se ha realizado sin necesidad de realizar ninguna prueba, simplemente deduciendo el incremento que supone la inclusión del MIC en la carga útil de las tramas. La Figura 39 ilustra el coste que supone la seguridad respecto al tamaño total de la trama. Los niveles NO_SECURITY y CTR no añaden información a las tramas, por lo que el impacto debido a la seguridad es nulo. En cambio, el resto de niveles suponen un coste mayor cuanto menor es la trama enviada.

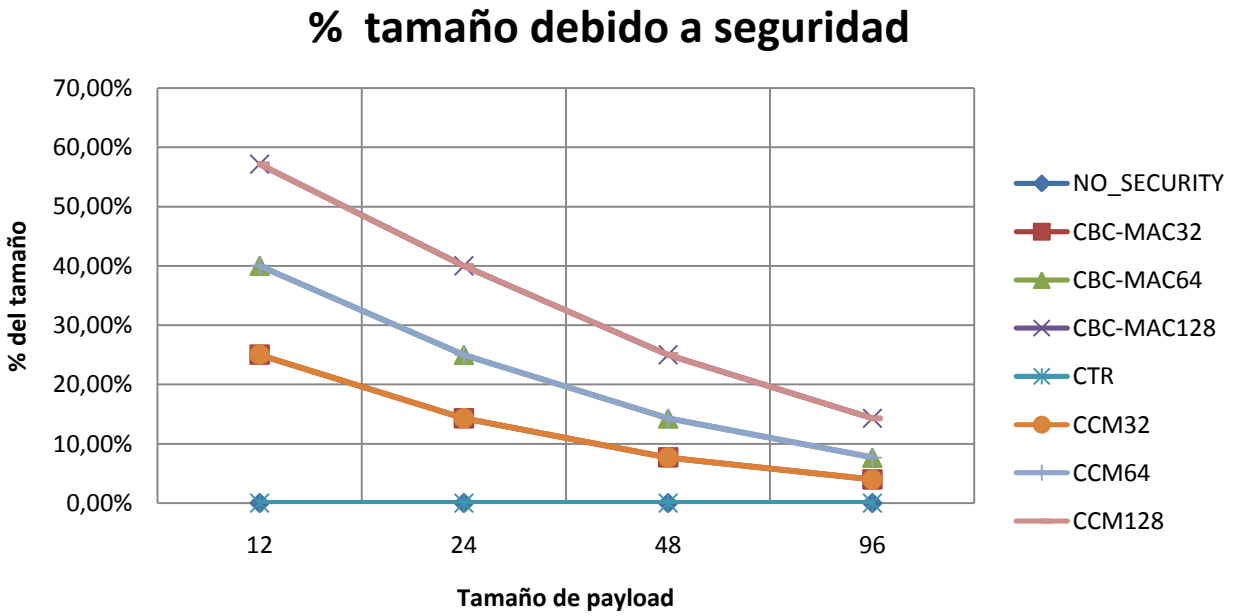

Figura 39 Impacto en el tamaño de trama

6.4.4. CUADRANTES

Este análisis enfrenta los diferentes niveles de seguridad frente a su coste en consumo y en tiempo de transmisión. De esta manera, aquellos que supongan menor coste se encontrarán ubicados en la parte inferior izquierda del cuadrante, mientras que aquellos que provocan un mayor impacto se colocarán en la esquina superior derecha.

El análisis incluye una cuantificación aproximada de la "cantidad" de seguridad de cada nivel de seguridad. De esta manera, CBC-MAC aportará menor seguridad que CCM, ya que este último aporta, además de la autenticación, medidas para preservar la confidencialidad. Se ha seguido el esquema seguido por el estándar IEEE 802.15.4, ilustrado en la Tabla 3. En el gráfico se representa la seguridad mediante el área de las esferas, cuanto mayor sea, más seguro se considera.

La aproximación dada en el estándar no deja de ser inexacta, ya que se comparan funcionalidades diferentes, no la fortaleza de un algoritmo o la resistencia al criptoanálisis. De hecho, no contar con un mecanismo que asegure la integridad conlleva un riesgo muy alto que no solventa el cifrado de las tramas.

Se han obtenido los cuadrantes para tamaños de trama de 12 bytes (Figura 40) y 96 bytes (Figura 41), respectivamente. Como se observa en las figuras, la posición relativa de las esferas apenas varía.

Es necesario destacar que las esferas se alinean respecto a una línea diagonal. Esto es debido a que los valores de consumo y de tiempo de transmisión son directamente proporcionales, pues al aumentar el tiempo de transmisión, se incrementa el consumo de energía.

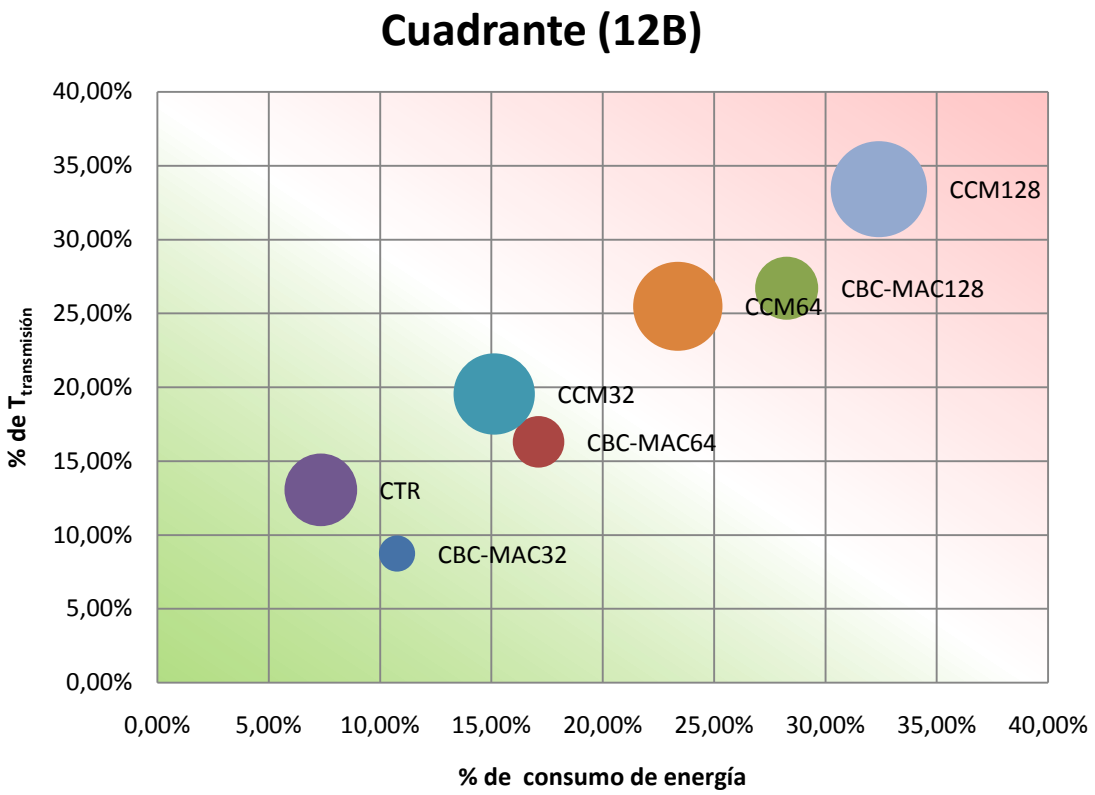

Figura 40 Cuadrante para tamaños de 12 bytes

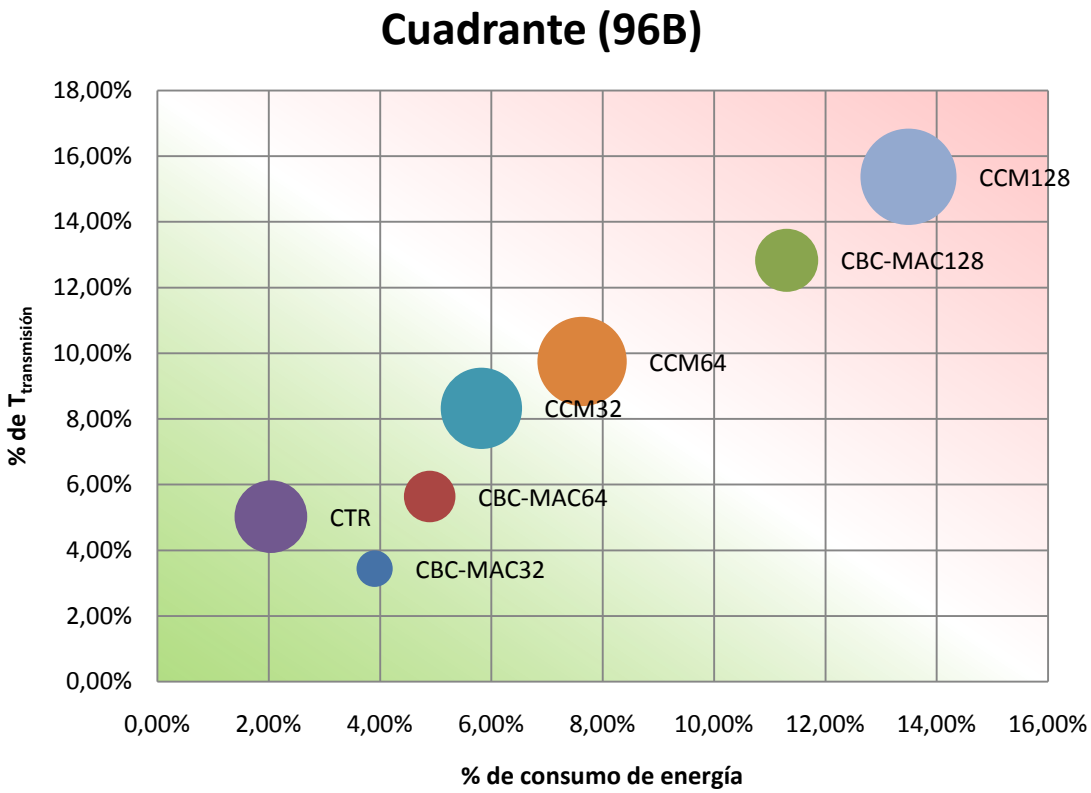

Figura 41 Cuadrante para tamaños de 96 bytes

7. CONCLUSIONES

7.1. CONCLUSIONES DEL ESTUDIO

Como se ha comprobado en los resultados del estudio, la seguridad supone un sobrecoste moderado en el consumo de energía de las redes de sensores, y presenta una característica común en todos los modos. Los mecanismos de seguridad que añaden información a las tramas debido a la autenticación, evidencian un consumo de energía superior, precisamente por el uso de espacio extra. Esta carga adicional provoca que los tiempos de transmisión de las tramas aumenten, incrementando el consumo de energía. De hecho, el consumo de energía debido al uso de la radio es mucho más significativo que el debido al módulo criptográfico.

Por tanto, los factores que provocan el aumento en el consumo son dos: el aumento del tiempo de transmisión provocado por el aumento en el tamaño de la trama, y en menor medida, el tiempo de *setup* del módulo criptográfico.

Sin embargo, este consumo se reduce a medida que el tamaño de las tramas se acerca al tamaño máximo. Teniendo en cuenta que uno de los objetivos de estas redes es reducir el número de transmisiones, mediante la agregación de la información, la longitud de trama típica será igual o cercana a la máxima. Es de esperar que en aplicaciones reales la seguridad suponga un impacto similar al resultado del análisis con tramas de 96 bytes.

Según los cuadrantes mostrados en el apartado 6.4.4, la habilitación del cifrado en las comunicaciones afecta ligeramente al consumo de los sensores. Vista la cantidad de ataques posibles a las redes de sensores por parte de nodos no autorizados, podría considerarse muy conveniente habilitar el modo CTR de cifrado, que sólo incrementa el consumo en un 7,4%, en el peor de los casos. Sin embargo, no habilitar controles de autenticación basados en criptografía supone un riesgo muy alto.

Como se comentaba en el apartado 4.3.3, el uso de cifrado CTR por sí solo está totalmente desaconsejado, ya que presenta vulnerabilidades e incurre en una falsa sensación de seguridad. Para considerar las comunicaciones totalmente seguras en términos de confidencialidad e integridad, se debe utilizar el modo CCM, con 32 bits para el mensaje de autenticación siempre que se cuente con mecanismos para renovar las claves periódicamente. El efecto en el consumo variará entre el 15% y el 6%, dependiendo del tamaño de la trama.

Sin embargo, hay que destacar que la seguridad debe ajustarse a las necesidades de las aplicaciones. Es necesario conocer qué necesita una aplicación y qué uso hace del canal de comunicación para poder determinar, junto con los resultados presentados en este estudio, el nivel de seguridad óptimo. Lo que se ha demostrado es que el coste de la seguridad es perfectamente asumible desde el punto de vista del consumo, del ancho de banda y del tiempo de transmisión en aplicaciones reales y con los dispositivos disponibles en el mercado.

7.2. LINEAS FUTURAS

Como se ha comentado a lo largo del estudio, a pesar de que la seguridad no implique demasiados sacrificios en cuanto al consumo de energía, y virtualmente ninguno en el consumo de CPU, los niveles de seguridad descritos en el estándar IEEE 802.15.4 no están libres de vulnerabilidades y fallos en el diseño. Es posible realizar ataques de denegación de servicio utilizando tramas especiales que invaliden las comunicaciones cifradas, impidiendo la comunicación entre nodos legítimos. Adicionalmente, si no se utilizan mecanismos que preserven la integridad de las tramas, aunque éstas viajen cifradas es posible manipularlas y provocar comportamientos erráticos y, potencialmente, facilitar ataques contra la privacidad.

Por otro lado, la implementación de este proyecto podría haberse completado con mecanismos de gestión de las claves, habilitando la renovación de las claves de cifrado de manera periódica, lo que permitiría evaluar los costes en memoria y CPU, pudiendo caracterizar de manera más fidedigna los costes globales en aplicaciones reales.

Según estas ideas, se proponen las siguientes líneas de investigación:

- Proponer e implementar mejoras sobre el modo CTR y el control de acceso a nodos no autorizados, mediante un proceso de asociación autenticado. Implementar un algoritmo de renovación de claves de red basado en criptografía simétrica, similar al propuesto en [32], utilizando el *hardware* del transceptor CC2420.

- Implementar una capa superior que permita gestionar claves de red y claves de sesión, pudiendo renovar las claves de manera periódica, definir claves para grupos de nodos, o incluso habilitar las comunicaciones privadas uno a uno. Analizar igualmente el consumo de potencia con las nuevas funcionalidades.

- Investigar la aplicación del cifrado asimétrico en las redes de sensores. El cifrado asimétrico resulta mucho más escalable que el cifrado simétrico, aunque es computacionalmente más costoso y requiere claves de mayor longitud para obtener el mismo nivel de seguridad. Las aproximaciones híbridas, basadas en el cifrado de curva elíptica para el intercambio de claves y cifrado de las comunicaciones mediante claves simétricas de sesión, pueden ser más efectivas en redes de sensores.

APÉNDICE

En este apéndice se adjuntan todas las tablas de resultados que se han obtenido durante la toma de medidas.

A. ENERGÍA CONSUMIDA

	Payload Bytes			
	12	24	48	96
NO_SECURITY	88,053 mJ	112,759 mJ	167,247 mJ	262,267 mJ
CBC-MAC32	98,669 mJ	124,252 mJ	173,460 mJ	272,921 mJ
CBC-MAC64	106,231 mJ	127,870 mJ	177,370 mJ	275,754 mJ
CBC-MAC128	122,731 mJ	147,336 mJ	197,139 mJ	295,691 mJ
CTR	95,020 mJ	120,077 mJ	169,280 mJ	267,731 mJ
CCM32	103,747 mJ	130,578 mJ	179,576 mJ	278,479 mJ
CCM64	114,905 mJ	138,648 mJ	187,834 mJ	283,930 mJ
CCM128	130,251 mJ	155,564 mJ	204,702 mJ	303,178 mJ

Tabla 4 Energía consumida

	Payload Bytes			
	12	24	48	96
CBC-MAC32	10,760%	9,250%	3,582%	3,904%
CBC-MAC64	17,112%	11,818%	5,707%	4,891%
CBC-MAC128	28,255%	23,468%	15,163%	11,304%
CTR	7,332%	6,094%	1,201%	2,041%
CCM32	15,127%	13,646%	6,866%	5,822%
CCM64	23,369%	18,673%	10,960%	7,630%
CCM128	32,397%	27,516%	18,298%	13,494%

Tabla 5 Porcentaje de consumo

B. TIEMPO DE TRANSMISIÓN

	Payload Bytes			
	12	24	48	96
NO_SECURITY	2,504 ms	3,168 ms	4,400 ms	6,960 ms
CBC-MAC32	2,744 ms	3,424 ms	4,688 ms	7,208 ms
CBC-MAC64	2,992 ms	3,568 ms	4,824 ms	7,376 ms
CBC-MAC128	3,416 ms	4,064 ms	5,360 ms	7,984 ms
CTR	2,880 ms	3,552 ms	4,760 ms	7,328 ms
CCM32	3,112 ms	3,776 ms	5,104 ms	7,592 ms
CCM64	3,360 ms	3,984 ms	5,224 ms	7,712 ms
CCM128	3,760 ms	4,432 ms	5,680 ms	8,224 ms

Tabla 6 Tiempo de transmisión

	Payload Bytes			
	12	24	48	96
CBC-MAC32	8,746%	7,477%	6,143%	3,441%
CBC-MAC64	16,310%	11,211%	8,789%	5,640%
CBC-MAC128	26,698%	22,047%	17,910%	12,826%
CTR	13,056%	10,811%	7,563%	5,022%
CCM32	19,537%	16,102%	13,793%	8,325%
CCM64	25,476%	20,482%	15,773%	9,751%
CCM128	33,404%	28,520%	22,535%	15,370%

Tabla 7 Porcentaje del tiempo de transmisión

C. TAMAÑO DE TRAMA

	Payload Bytes			
	12	24	48	96
NO_SECURITY	0,00%	0,00%	0,00%	0,00%
CBC-MAC32	25,00%	14,29%	7,69%	4,00%
CBC-MAC64	40,00%	25,00%	14,29%	7,69%
CBC-MAC128	57,14%	40,00%	25,00%	14,29%
CTR	0,00%	0,00%	0,00%	0,00%
CCM32	25,00%	14,29%	7,69%	4,00%
CCM64	40,00%	25,00%	14,29%	7,69%
CCM128	57,14%	40,00%	25,00%	14,29%

Tabla 8 Porcentaje del tamaño de trama

D. MEDIDAS

	Payload Bytes																			
	12					24					48					96				
	V_{ON}	V_{OFF}	I_{ON}	I_{OFF}	Tb	V_{ON}	V_{OFF}	I_{ON}	I_{OFF}	Tb	V_{ON}	V_{OFF}	I_{ON}	I_{OFF}	Tb	V_{ON}	V_{OFF}	I_{ON}	I_{OFF}	Tb
NO_SECURITY	144,890	48,000	11,145	3,692	2,504	146,740	48,000	11,288	3,692	3,168	157,230	48,000	12,095	3,692	4,400	155,800	48,000	11,985	3,692	6,960
CBC-MAC32	148,320	48,000	11,409	3,692	2,744	149,750	48,000	11,519	3,692	3,424	152,840	48,000	11,757	3,692	4,688	156,590	48,000	12,045	3,692	7,208
CBC-MAC64	146,360	48,000	11,258	3,692	2,992	147,800	48,000	11,369	3,692	3,568	151,830	48,000	11,679	3,692	4,824	154,510	48,000	11,885	3,692	7,376
CBC-MAC128	148,190	48,000	11,399	3,692	3,416	149,600	48,000	11,508	3,692	4,064	151,880	48,000	11,683	3,692	5,360	152,990	48,000	11,768	3,692	7,984
CTR	135,540	48,000	10,426	3,692	2,880	139,030	48,000	10,695	3,692	3,552	146,610	48,000	11,278	3,692	4,760	150,820	48,000	11,602	3,692	7,328
CCM32	137,020	48,000	10,540	3,692	3,112	142,370	48,000	10,952	3,692	3,776	144,970	48,000	11,152	3,692	5,104	151,450	48,000	11,650	3,692	7,592
CCM64	140,720	48,000	10,825	3,692	3,360	143,320	48,000	11,025	3,692	3,984	148,310	48,000	11,408	3,692	5,224	152,040	48,000	11,695	3,692	7,712
CCM128	142,630	48,000	10,972	3,692	3,760	144,610	48,000	11,124	3,692	4,432	148,670	48,000	11,436	3,692	5,680	152,250	48,000	11,712	3,692	8,224

Tabla 9 Medidas de las pruebas

- V_{ON}: Voltaje medio durante el tiempo de transmisión
- V_{OFF}: Voltaje medio entre la llegada del *beacon* y el inicio de la transmisión
- I_{ON}: Intensidad media durante el tiempo de transmisión.
- I_{OFF}: Intensidad media entre la llegada del *beacon* y el inicio de la transmisión
- T_b: Tiempo desde el inicio de la transmisión USART hasta el final de la transmisión RF.

E. CÁLCULOS INTERMEDIOS

	Payload Bytes							
	12				24			
	$Consumo_{ON}$ (mW)	$Consumo_{OFF}$ (mW)	Tiempo (ms)	% overhead	$Consumo_{ON}$ (mW)	$Consumo_{OFF}$ (mW)	Tiempo (ms)	% overhead
NO_SECURITY	35,165	12,007	2,504	0,00%	35,593	12,007	3,168	0,00%
CBC-MAC32	35,958	12,007	2,744	25,00%	36,288	12,007	3,424	14,29%
CBC-MAC64	35,505	12,007	2,992	40,00%	35,838	12,007	3,568	25,00%
CBC-MAC128	35,928	12,007	3,416	57,14%	36,254	12,007	4,064	40,00%
CTR	32,993	12,007	2,880	0,00%	33,805	12,007	3,552	0,00%
CCM32	33,338	12,007	3,112	25,00%	34,581	12,007	3,776	14,29%
CCM64	34,198	12,007	3,360	40,00%	34,801	12,007	3,984	25,00%
CCM128	34,641	12,007	3,760	57,14%	35,100	12,007	4,432	40,00%

	Payload Bytes							
	48				96			
	$Consumo_{ON}$ (mW)	$Consumo_{OFF}$ (mW)	Tiempo (ms)	% overhead	$Consumo_{ON}$ (mW)	$Consumo_{OFF}$ (mW)	Tiempo (ms)	% overhead
NO_SECURITY	38,011	12,007	4,400	0,00%	37,682	12,007	6,960	0,00%
CBC-MAC32	37,001	12,007	4,688	7,69%	37,864	12,007	7,208	4,00%
CBC-MAC64	36,768	12,007	4,824	14,29%	37,385	12,007	7,376	7,69%
CBC-MAC128	36,780	12,007	5,360	25,00%	37,035	12,007	7,984	14,29%
CTR	35,563	12,007	4,760	0,00%	36,535	12,007	7,328	0,00%
CCM32	35,183	12,007	5,104	7,69%	36,681	12,007	7,592	4,00%
CCM64	35,956	12,007	5,224	14,29%	36,817	12,007	7,712	7,69%
CCM128	36,039	12,007	5,680	25,00%	36,865	12,007	8,224	14,29%

Tabla 10 Cálculos intermedios

- $Consumo_{ON}$: Consumo de energía medio durante el tiempo de transmisión.
- $Consumo_{OFF}$: Consumo medio de energía entre la llegada del *beacon* y el inicio de la transmisión.
- Tiempo: Tiempo de transmisión.
- %overhead: Porcentaje que supone el MIC respecto al tamaño del *payload.*

BIBLIOGRAFÍA

[1] REALTIME, "SHIMMER Workshop Manual," ed, September 2008.

[2] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," Proceedings of the IEEE, vol. 91, pp. 1247-1256, 2003.

[3] Á. Lédeczi, et al., "Countersniper system for urban warfare," vol. 1, p. 157, 2005.

[4] Zigbee Alliance. Available: http://www.zigbee.org

[5] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," IEEE Std 802.15.4-2003, pp. 0_1-670, 2003.

[6] "IEEE Standard for Information Technology- Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)," IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 0_1-305, 2006.

[7] Directiva 95/46/CE del Parlamento Europeo y del Consejo. Available: http://europa.eu/legislation_summaries/information_society/l14012_es.htm

[8] Staff Discussion Draft for a Bill to require notice to and consent of an individual prior to the Collection and disclosure of certain personal information relating to that individual. Available: http://www.boucher.house.gov/images/stories/Privacy_Draft_5-10.pdf

[9] M. Saraogi, "Security in Wireless Sensor Networks," ed.

[10] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on, 2003, pp. 113-127.

[11] J. P. Walters, et al., "Wireless sensor network security: A survey," in book chapter of Security," ed. in Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds: CRC Press, 2007.

[12] F. Stajano and R. Anderson, "The Resurrecting Duckling: security issues for ubiquitous computing," Computer, vol. 35, pp. 22-26, 2002.

[13] N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," presented at the Proceedings of the 3rd ACM workshop on Wireless security, Philadelphia, PA, USA, 2004.

[14] TI. (July 2009). MSP430 Memory Programming User Guide. Available: http://www.ti.com/litv/pdf/slau265i

[15] A. Becher, et al., "Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks," presented at the SPC, 2006.

[16] I. Shin, et al., "Reactive jamming attacks in multi-radio wireless sensor networks: an efficient mitigating measure by identifying trigger nodes," presented at the Proceedings of the 2nd ACM international workshop on Foundations of wireless ad hoc and sensor networking and computing, New Orleans, Louisiana, USA, 2009.

[17] Z. Qinghua, et al., "Defending against Sybil attacks in sensor networks," in Distributed Computing Systems Workshops, 2005. 25th IEEE International Conference on, 2005, pp. 185-191.

[18] Y.-C. Hu, et al., "Packet leashes: a defense against wormhole attacks in wireless ad hoc networks," ed. Proceedings of IEEE NFOCOM, 2003.

[19] D. Gascón. (February 2009). Security in 802.15.4 and ZigBee networks. Available: http://www.sensor-networks.org

[20] Y. Xiao, et al., "MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks," EURASIP J. Wirel. Commun. Netw., vol. 2006, pp. 81-81, 2006.

[21] NIST. (November 2001). FIPS PUB 197, Advanced Encryption Standard (AES). Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[22] NSA. (June 2003). National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information. Available: http://www.cnss.gov/Assets/pdf/cnssp_15_fs.pdf

[23] E. Barker, et al. (2002). Recommendation for Key Management – Part 2: Best Practices for Key Management Organization. Available: http://csrc.nist.gov/publications/nistpubs/800-57/SP800-57-Part2.pdf

[24] D. Whiting, et al. (September 2003). RFC 3610: Counter with CBC-MAC (CCM). Available: http://tools.ietf.org/rfc/rfc3610.txt

[25] "SHIMMER Workshop Manual," in Sensing Health with Intelligence Modularity, Mobility and Experimental Reusability, ed, September 2008.

[26] A. Cunha, et al., "IPP Hurray. An IEEE 802.15.4 protocol implementation (in nesC/TinyOS)," ed: Polytechnic Institute of Porto (ISEP-IPP), May 2007.

[27] (2004). Chipcon CC2420 Data Sheet. Available: http://www.chipcon.com/files/CC2420_Data_Sheet_1_1.pdf

[28] M. Healy and et al., "Efficiently securing data on a wireless sensor network," Journal of Physics: Conference Series, vol. 76, p. 012063, 2007.

[29] M. T. Hansen, "Asynchronous group key distribution on top of the cc2420 security mechanisms for sensor networks," presented at the Proceedings of the second ACM conference on Wireless network security, Zurich, Switzerland, 2009.

[30] W. Diffie and M. E. Hellman, "Multiuser cryptographic techniques," presented at the Proceedings of the June 7-10, 1976, national computer conference and exposition, New York, New York, 1976.

[31] S. Khajuria and H. Tange, "Implementation of Diffie-Hellman key exchange on wireless sensor using elliptic curve cryptography," in Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology, 2009. Wireless VITAE 2009. 1st International Conference on, 2009, pp. 772-776.

[32] J. Schaad and R. Housley. (September 2002). RFC 3394: Advanced Encryption Standard (AES) Key Wrap Algorithm. Available: http://www.ietf.org/rfc/rfc3394.txt